NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

JPEG2000 IMAGE COMPRESSION AND ERROR RESILIENCE FOR TRANSMISSION OVER WIRELESS CHANNELS

by

Konstantinos Kamaras

March 2002

Thesis Advisor:	Murali Tummala
Second Reader:	Robert Ives

Report Documentation Page

Report Date	Report Type	Dates Covered (from... to)
29 Mar 2002	N/A	-

Title and Subtitle JPEG2000 Image Compression and Error Resilence for Transmission over Wireless Channels	**Contract Number**
	Grant Number
	Program Element Number
Author(s) Kamaras, Konstantinos	**Project Number**
	Task Number
	Work Unit Number
Performing Organization Name(s) and Address(es) Naval Postgradaute School Monterey, California	**Performing Organization Report Number**
Sponsoring/Monitoring Agency Name(s) and Address(es)	**Sponsor/Monitor's Acronym(s)**
	Sponsor/Monitor's Report Number(s)
Distribution/Availability Statement Approved for public release, distribution unlimited	
Supplementary Notes The original document contains color images.	
Abstract	
Subject Terms	
Report Classification unclassified	**Classification of this page** unclassified
Classification of Abstract unclassified	**Limitation of Abstract** UU
Number of Pages 121	

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			*Form Approved OMB No. 0704-0188*
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.			
1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** March 2002	**3. REPORT TYPE AND DATES COVERED** Master's Thesis	
4. TITLE AND SUBTITLE: JPEG2000 Image Compression and Error Resilience For Transmission Over Wireless Channels			**5. FUNDING NUMBERS**
6. AUTHOR(S) Konstantinos Kamaras			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000			**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A			**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.			
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited			**12b. DISTRIBUTION CODE**

13. ABSTRACT *(maximum 200 words)*

This thesis examines the compression performance of the JPEG2000 standard for image transmission over noisy channels. Other features of the standard, such as error resilience and region of interest, have been studied and their effectiveness tested on several images. The JPEG2000 still image compression standard has provided higher compressions performance with lower distortion and better image quality than JPEG. JPEG2000 has the capability to define regions of interest of any shape and size and code the selected regions with a higher fidelity than the rest of the image. Compressed image data is transmitted over a noisy wireless channel based on Gilbert-Eliot model, which simulates both isolated and burst errors. JPEG2000 error resilient tools are used to allow the decoder to detect and conceal errors introduced in the channel. The results indicate up to 10 dB improvement in the peak signal to noise ratio when these tools are used

14. SUBJECT TERMS Wavelet Analysis, Discrete Wavelet Transform, JPEG2000, Forward Error Correction (FEC), Automatic Repeat Request (ARQ), Markov Channel Model			**15. NUMBER OF PAGES** 121
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UL

NSN 7540-01-280-5500 Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

JPEG2000 IMAGE COMPRESSION AND ERROR RESILIENCE FOR TRANSMISSION OVER WIRELESS CHANNELS

Konstantinos Kamaras
Lieutenant, Hellenic Navy
B.S., Hellenic Naval Academy, 1993

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
March 2002

Author: Konstantinos Kamaras

Approved by: Murali Tummala
Thesis Advisor

Robert Ives
Second Reader

Jeffrey B. Knorr
Chairman, Department of Electrical and Computer Engineering

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

This thesis examines the compression performance of the JPEG2000 standard for image transmission over noisy channels. Other features of the standard, such as error resilience and region of interest, have been studied and their effectiveness tested on several images. The JPEG2000 still image compression standard has provided higher compressions performance with lower distortion and better image quality than JPEG. JPEG2000 has the capability to define regions of interest of any shape and size and code the selected regions with a higher fidelity than the rest of the image. Compressed image data is transmitted over a noisy wireless channel based on Gilbert-Eliot model, which simulates both isolated and burst errors. JPEG2000 error resilient tools are used to allow the decoder to detect and conceal errors introduced in the channel. The results indicate up to 10 dB improvement in the peak signal to noise ratio when these tools are used.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ABBREVIATIONS

ARQ	Automatic Repeat Request
BER	Bit Error Rate
BSC	Binary Symmetric Channel
DCT	Discrete Cosine Transform
DWT	Discrete Wavelet Transform
EBCOT	Embedded Block Coding with Optimized Truncation
FEC	Forward Error Correction
FT	Fourier Transform
HARQ	Hybrid Automatic Repeat Request
IFT	Inverse Fourier Transform
ISO	International Standardization Organization
ITU	International Telecommunication Union
JPEG	Joint Photographic Expert Group
LSB	Least Significant Bit
MRA	Multiresolution Analysis
MSB	Most Significant Bit
PSNR	Peak-Signal-to-Noise-Ratio
QMF	Quadtrature Mirror Filters
SQ	Scalar Quatization
STFT	Sort Time Fourier Transform
TCQ	Trellis Coded Quantization

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

A Master of Science in Electrical Engineering was a challenge for my abilities. My achievement would not have happened without the people that I have around me everyday of my life. To my wife Eleni I would like to dedicate this work that took a year to complete. Her strength and support enabled me to work hard to achieve this goal. To my son Raphelo I would like to dedicate my diploma. He brings joy into our every day life and he has taught us much about the priorities of our life.

I would like to thank Professors Murali Tummala and Robert Ives. With their professional guidance and passion about new technology, they were a constant support throughout my thesis research and writing.

Finally, I would like to thank Professor A. N. Skodras from the University of Patras, Greece and a member of the JPEG committee, who provide me with the latest reference manual of the JPEG2000 VM8.5 source code.

THIS PAGE INTENTIONALLY LEFT BLANK

EXECUTIVE SUMMARY

The demand for image compression performance superior to the existing standards and the need for robust image transmission over wireless channels have increased in the last years due to the explosive growth of network applications and mobile multimedia. A new still image compression standard, known as JPEG2000, is designed to compliment the existing JPEG. JPEG2000 is a wavelet-based codec, which supports different types of still images and provides tools for a wide variety of applications, such as Internet, image library, and real-time transmission through wireless channels.

This thesis investigates the compression performance of the JPEG2000 standard, in comparison with JPEG, for image transmission over wireless bandlimited noisy channels. Other features of the standard, such as error resilience and region of interest, have also been studied and their effectiveness tested on several images. The thesis also examines the effect of channel coding techniques, such as forward error correction, automatic repeat request (ARQ) and hybrid-ARQ, in combination with JPEG2000's error resilient tools on the perceived quality of the image after transmission through an unreliable channel. The communication channel used is based upon the Gilbert-Eliot model with an embedded two-state Markov process for simulating slow fading conditions.

JPEG2000 still image compression standard has provided high compression rates (better than 80:1) with low distortion, and image quality significantly better than JPEG. While its performance is superior to that of JPEG, the JPEG2000 algorithm is more complex and computationally more expensive than JPEG.

Image compression with a specified region of interest using JPEG2000 has also been examined. This feature of JPEG2000 enables the user to define regions of interest of any shape and size and code the selected regions at a better quality than the rest of the image. The effectiveness of the region of interest feature is demonstrated using several images and for different shapes.

Both compression schemes have been investigated for image transmission over bandwidth-limited, noisy channels. The bitstreams of both JPEG and JPEG2000 were encoded using three different error control schemes: convolutional forward error correction code, stop-and–wait ARQ and hybrid-ARQ. Based on simulation results, baseline JPEG was found to be unreliable for image transmission over noisy channels due to frequent loss of synchronization between the bitstream and the decoder. In comparison, JPEG2000 provides various error resilient mechanisms that enable the decoder not only to achieve synchronization with the bitstream, but also to detect and correct errors that were injected into the bitstream during transmission. The results indicate up to 10 dB improvement in the peak signal-to-noise ratio of the received images when these tools are used.

An interesting extension of this work may consider more accurate channel models, such as a model based on a four-state Markov chain. Additional future effort may consider transmission of JPEG2000 bitstream enhanced with other forward error correction codes, such as turbo codes or Reed-Solomon codes. A simulation of a multi-node network for compressed image transmission along with the error resilient tools of JPEG2000 under network congestion conditions would be of interest.

I. INTRODUCTION

The demand for image compression performance superior to the existing standards and the need for robust image transmission over wireless channels have increased in the last few years due to the explosive growth of network applications and mobile multimedia. The Joint Photographic Expert Group 2000 (JPEG2000), which is both an ITU-T standard (ITU0T.800) and an ISO standard (ISO 15444), addresses the compression of a wide spectrum of still images and provides reliable error resilient bitstream coding for transmission through unreliable channels.

JPEG2000 is a wavelet-based codec and is intended to support different types of still images, such as bi-level, gray-level and multi-component, with different characteristics such as natural, scientific, medical, and text. JPEG2000 allows different image models including Internet applications, image library, and real-time transmission through channels with limited bandwidth. The new standard is designed to compliment the existing JPEG standard rather than replace it. Its coding system provides low bit-rate operation suitable for limited bandwidth networks, with low rate-distortion and subjective image quality superior to the currently used standard [1].

JPEG2000 uses the Embedded Block Coding with Optimized Truncation (EBCOT) to generate the embedded bit stream [1]. The main advantage of this algorithm is that the image need not be compressed multiple times in order to achieve the desired bit-rate, unlike with the JPEG compression standard. Another related advantage of practical significance is that the bitstream produced with this algorithm provides error resilient tools that allow the decoder to efficiently detect and to some extent undo the effect of error injections due to a noisy channel over which the image was transmitted.

A. THESIS OBJECTIVES

Given the importance of network image applications within the context of the present and future commercial and military systems, this thesis investigates the performance of the JPEG2000 compression standard and its error resilient mechanisms. The compression performance of JPEG2000 is evaluated and compared with that of the JPEG standard for different images. Different modes of compression provided by

JPEG2000 are tested and conclusions related to the performance are provided. The effectiveness of the region of interest coding is studied as well.

The thesis also examines the effect of channel coding techniques, such as Forward Error Correction (FEC), Automatic Repeat Request (ARQ) and Hybrid ARQ, in combination with JPEG2000's error resilient tools on the perceived quality of the image transmitted through an unreliable channel. The communication channel used is based upon the Gilbert-Eliot model with an embedded two-state Markov process for simulating slow fading conditions.

B. THESIS ORGANIZATION

This thesis is organized into five chapters and three supporting appendices. Chapter II provides an overview of wavelet theory and wavelet analysis. Chapter III describes the Joint Photographic Expert Group 2000 (JPEG2000) standard for still image compression. Also, tools that provide error resilience for JPEG2000 image transmission over error-prone channels are briefly described. Chapter IV presents the transmission schemes that will be used for the evaluation of image compression over unreliable media. The channel model used in simulations is also presented. Simulation results of JPEG2000 image transmission over unreliable channels and the application of optional error resilient tools to enhance image quality are presented. Chapter V summarizes the work and provides conclusions. Appendix A presents additional examples of comparison between JPEG2000 and JPEG. Appendix B includes results of image transmission for different channel conditions in addition to those presented in Chapter IV. Appendix C provides usage and examples of JPEG2000 VM8.5 source code.

II. INTRODUCTION TO WAVELETS AND THE WAVELET TRANSFORM

A. INTRODUCTION

Wavelet representation of signals provides a more efficient localization in both time and frequency or scale than the Fourier transform or the short-time Fourier transform. Wavelets also lend themselves to *multiresolution* analysis in which the signal is decomposed in terms of detail and approximation (resolution) coefficients. The multiresolution decomposition separates the components of a signal in such a way that it is more flexible than most other methods of analysis, processing, or compression. This approach can be used for linear as well as non-linear processing of signals and offers new methods for signal detection [2], classification [2], filtering [3], [2] and compression [4], [5].

This chapter introduces wavelets in order to understand how wavelet analysis is implemented in the JPEG2000 still image compression standard, presented in Chapter III. It provides examples of methods based on wavelet analysis in terms of one-dimensional signals and then extends them to image processing. MATLAB is the programming software used for most of the examples and figures presented.

B. SIGNAL TRANSFORMATIONS

The Fourier transform of a signal $x(t)$ is given by

$$X(f)=\int_{-\infty}^{+\infty} x(t)e^{-2j\pi ft}dt \tag{2.1}$$

and its inverse transform is

$$x(t)=\int_{-\infty}^{+\infty} X(f)e^{2j\pi ft}df \tag{2.2}$$

Although Fourier transform is widely used, it does not provide a desirable time-frequency representation when the signal is *non-stationary* [5]. One approach to this problem is to use the short-time Fourier transform, which assumes that short segments of *non-stationary* signals are stationary. A window function $W(t)$ is chosen to represent the segment over which the *stationarity* of the signal is valid. The mathematical expression for short-time Fourier transform is

$$\mathrm{X}(t,f)=\int_t \left[x(t)W^*(t-t')\right]e^{-2j\pi ft}dt \tag{2.3}$$

where the asterisk on $W^*(t)$ indicates complex conjugation. The short-time Fourier transform provides a true time-frequency representation of the signal.

Due to the *uncertainty principle*, the short-time Fourier transform can estimate the time intervals in which certain bands of frequencies exist but not the exact time a frequency of a signal changes. The time-frequency representation using the wavelet transform, on the other hand, has the capability to provide an exact time frequency characterization.

1. Wavelet Transform

The signal $x(t)$ can be represented as a linear combination

$$x(t)=\sum_l w_l \psi_l(t) \tag{2.4}$$

where w_l are the real-valued expansion coefficients and $\psi_l(t)$ are a set of real-valued functions of t [4]. The reader may note that the representation of Equation (2.4) is similar to that of the Fourier series. The set of functions $\psi_l(t)$ that uniquely represent a signal are referred to as a basis set.

From a function called the *mother wavelet*, a basis set can be realized through scaling and translation. The scaled and translated version of a mother wavelet function can be expressed as [4]

$$\psi_{\alpha,b}(t)=\frac{1}{\sqrt{\alpha}}\psi\left(\frac{t-b}{\alpha}\right) \tag{2.5}$$

where α and b represent the scale and translation parameters, respectively. Figure 2-1 shows the Meyer mother wavelet as well as its translated and scaled versions. Both translation and scale parameters are set equal to 2.

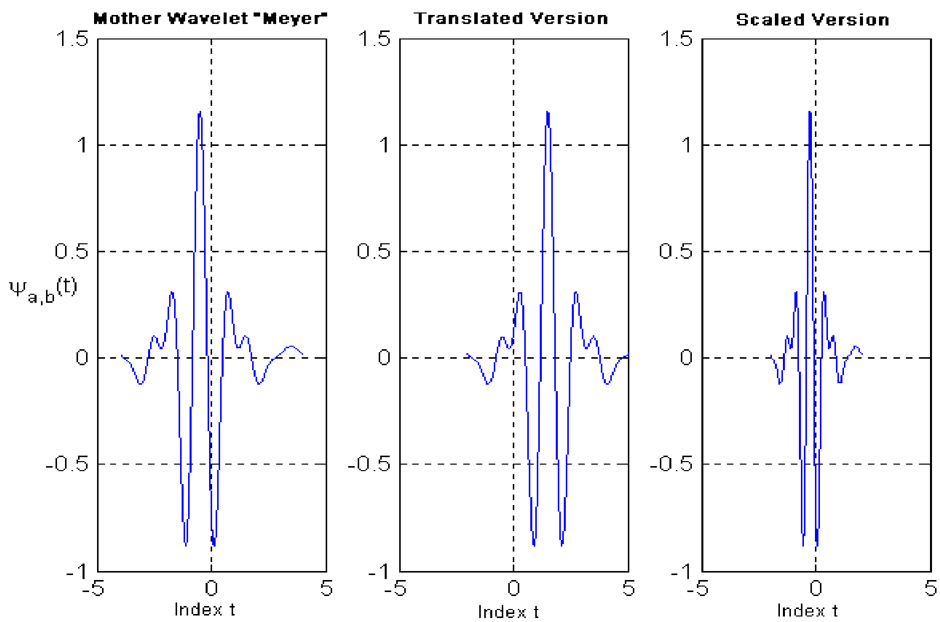


Figure 2-1. Meyer Mother Wavelet and its Translated and Scaled Versions.

The expansion coefficients of Equation (2.4) can also be denoted as a continuous valued wavelet transform $w_{\alpha,b}$ as given by the inner product [4]

$$w_{\alpha,b} = \langle\psi_{\alpha,b}(t), x(t)\rangle = \int_{-\infty}^{+\infty} \psi_{\alpha,b}(t)\, x(t)\, dt \tag{2.6}$$

where α and b are continuous valued parameters. The signal then can be represented as [4]

$$x(t) = \frac{1}{C_\psi} \int_{-\infty}^{+\infty} w_{\alpha,b} \psi_{\alpha,b}(t) \frac{d\alpha\ db}{\alpha^2} \tag{2.7}$$

where $C_\psi = \int_0^{+\infty} \frac{|\psi(\omega)|^2}{\omega} d\omega$ and $\Psi(\omega) = FT(\psi(t))$.

2. Discrete Wavelet Transform

As with other signal transforms, it is desirable to discretize the wavelet transform by discretizing the scaling and translation parameters α and b. A commonly used way to discretize α and b is as follows [5]

$$\alpha = \alpha_0^{-j}, \qquad b = kb_0\alpha^{-j} \qquad j,k \in \mathbb{Z} \tag{2.8}$$

where $\mathbb{Z}$ represents all the integer numbers. The most common choices for a_0 and b_0 are 2 and 1, respectively. This results in a two-dimensional representation of the wavelet. The value of parameter k represents the parameterization of time or space, and the value of j the frequency or the logarithm of scale. The discrete version of the wavelet set of Equation (2.5) is then given by [4], [5]

$$\psi_{j,k}\left(t\right) = 2^{j/2}\psi\left(2^j t - k\right), \quad j,k \in \mathbb{Z} \tag{2.9}$$

The function $x\left(t\right)$ can now be reconstructed from the wavelet coefficients as follows [4]

$$x\left(t\right) = \sum_k \sum_j w_{j,k}\psi_{j,k}\left(t\right) \tag{2.10}$$

where the wavelet coefficients $w_{j,k}$ are given by

$$w_{j,k} = \ < x(t), \psi_{j,k}\left(t\right) > \ = 2^{j/2}\int x\left(t\right)\psi\left(2^j t - k\right)dt. \tag{2.11}$$

C. MULTIRESOLUTION ANALYSIS

Let S be a vector space of signals. If $x\left(t\right) \in S$, then it can be expressed as [4]

$$x\left(t\right) = \sum_k a_k \phi_k\left(t\right) \tag{2.12}$$

where $\phi_k\left(t\right)$ form the basis set for space S for unique representation and can be obtained from the scaling function $\phi\left(t\right)$ as follows [4]

$$\phi_k(t) = \phi(t-k) \qquad k \in \mathbb{Z}\,,\ \phi \in L^2 \tag{2.13}$$

A two-dimensional expression of Equation (2.13), similar to that of Equation (2.9), is desirable [5]:

$$\phi_{j,k}(t) = 2^{j/2}\phi\left(2^j t - k\right) \tag{2.14}$$

where as in Equation (2.9) of the wavelet function, k represents the translation and j the scale. This two-dimensional family of functions, generated from the basic scaling function by scaling and translation, spans over k

$$V_j = \overline{\underset{k}{span}\left\{\phi_k\left(2^j t\right)\right\}} = \overline{\underset{k}{span}\left\{\phi_{j,k}(t)\right\}} \tag{2.15}$$

where V_j denotes the nesting of spans of $\phi\left(2^j t - k\right)$ and is graphically illustrated in Figure 2-2. Consequently, $V_0 \subset V_1 \subset V_2 ... \subset V_j \subset V_{j+1}$, $\forall j \in \mathbb{Z}$ [4]. From Figure 2-2, as the resolution increases to V_∞, the approximation signal converges to the original [6]. Subspaces W_j are explained in the next subsection.

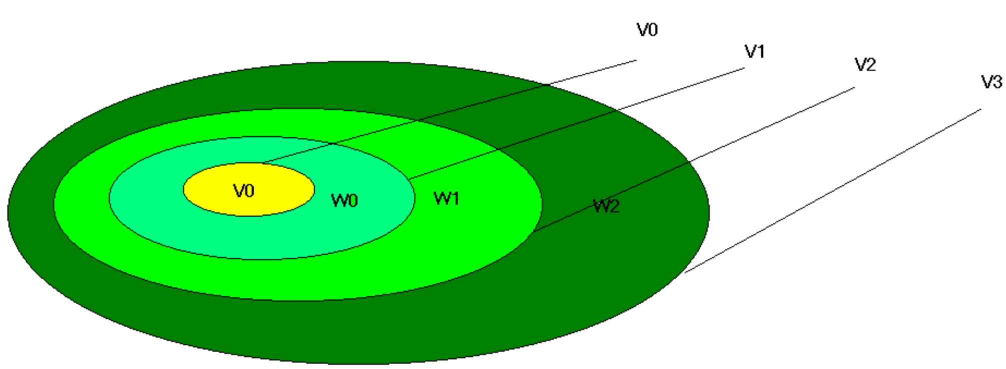


Figure 2-2. Nested Vector Spaces Spanned by the Scaling Functions from [4].

Also, the nested vector spaces must satisfy the following scaling condition

$$\phi(t) \in V_j \Leftrightarrow \phi(2t) \in V_{j+1} \tag{2.16}$$

This condition allows $\phi(t)$ to be expressed as a weighted sum of shifted $\phi(2t)$ as

$$\phi(t) = \sum_n h(n)\sqrt{2}\phi(2t-n), \quad n \in \mathbb{Z} \tag{2.17}$$

where $h(n)$ is a sequence of real or complex numbers called scaling function coefficients. This is the *multiresolution analysis* (MRA) equation or the *dilation equation* [4].

Figure 2-3 shows how scaling functions can be used to approximate an envelope of speech signal. The first plot is the original signal and the following plots are the approximations of the signal after projection onto subspaces V_0, V_1, V_2 and V_3. As can be seen, by moving to a higher resolution vector space, the approximation of the signal becomes better. The subscript j of $\phi_j(t)$ represents the scale of the basis function. For this example, the basis function is Daubechies'3 (db3).

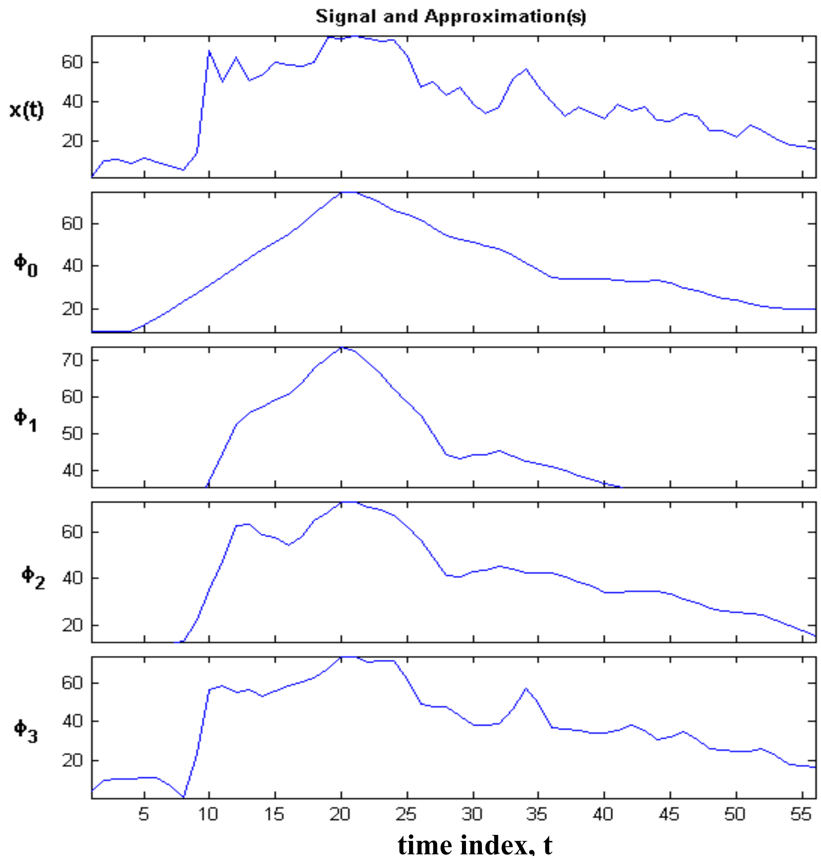


Figure 2-3. Approximation of a Signal $x(t)$ by using the Scaling Function Daubechies'3 or db3.

1. Wavelet Vector Space and Wavelet Function

The best way to represent a signal $x(t)$ is not by using a set of scaling functions at multiple resolution subspaces, but by defining a set of functions that span the differences between subspaces $V_0, V_1, V_2, \ldots, \text{etc}$. These functions are the wavelets $\psi_{j,k}(t)$ as defined in Equation (2.9). The subspaces spanned by the wavelet functions are denoted as W_j and are orthogonal to V_j. The scaling and wavelet spaces satisfy the following relationship

$$V_{j+1} = V_j \oplus W_j \tag{2.18}$$

which is illustrated in Figure 2-2.

Similar to Equation (2.17), a mother wavelet function can be represented as a weighted sum of scaled and translated functions

$$\psi(t) = \sum_{n} w(n)\sqrt{2}\phi(2t-n), \quad n \in \mathbb{Z} \tag{2.19}$$

where $w(n)$ is a set of expansion coefficients. Equation (2.19) is a fundamental wavelet equation and will be used later for implementing the wavelet multiresolution analysis.

D. IMPLEMENTATION OF MRA

1. Wavelet Representation of Signal

The signal $x(t)$ can be represented as [4]

$$x(t) = \sum_{k} c_{j_0,k} 2^{j_0/2}\phi\left(2^{j_0}t-k\right) + \sum_{k}\sum_{j=j_0}^{\infty} d_{j,k} 2^{j/2}\psi\left(2^{j}t-k\right) \tag{2.20}$$

where j_0 can take any value depending on the resolution level to which the representation corresponds. The coefficients $c_{j,k}$ and $d_{j,k}$ are the scaling and wavelet coefficients, respectively, and are defined by [5]

$$c_{j,k} = \; < x(t), \phi_{j,k}(t) > \; = \sum_{m} h(m-2k)c_{j+1}(m) \tag{2.21}$$

$$d_{j,k} = \; < x(t), \psi_{j,k}(t) > \; = \sum_{m} w(m-2k)c_{j+1}(m) \tag{2.22}$$

where $m = 2k+n$. The first term, a sum of all the scaling functions at scale j_0 for all translations, will give the approximation of $x(t)$. The second term, a double sum of all the scales of the wavelet function starting from j_0 for all translations, will give the details.

Figures 2-4 and 2-5 illustrate the wavelet decomposition of a signal by showing the components of the signal that exist in the wavelet spaces W_j at different scales j. Figure 2-4 shows a signal with a discontinuity. Figure 2-5 shows the wavelet decomposition of this signal. The scaling function $\phi_0(t)$ by itself can approximate the signal but it is not able to preserve the discontinuity. However, as we move to higher resolutions, we observe that the discontinuity is isolated and located by the wavelet function $\psi_j(t)$.

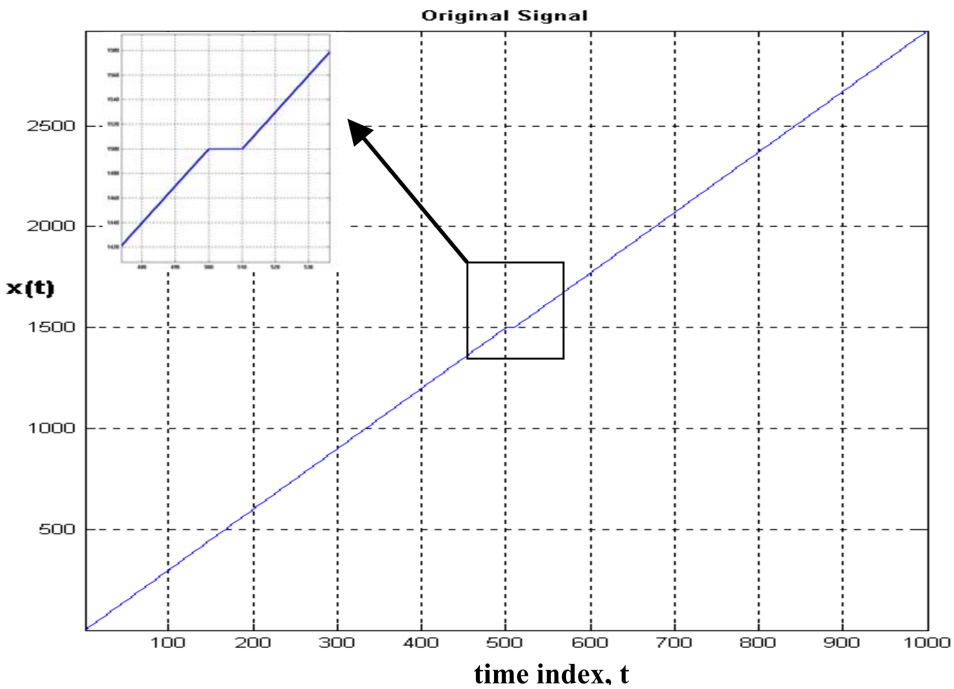


Figure 2-4. A Signal with Discontinuity.

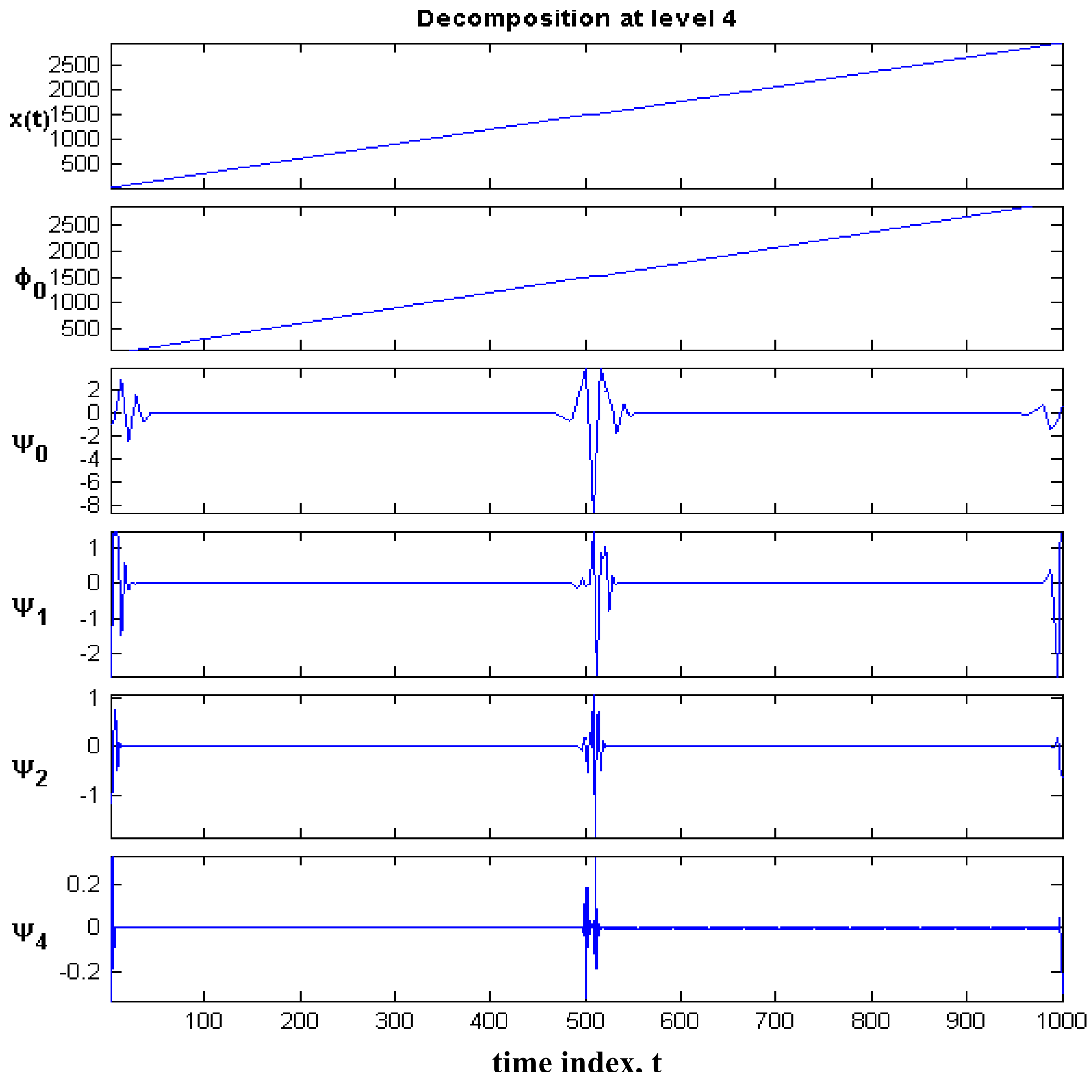


Figure 2-5. Wavelet Analysis of a Signal with Discontinuity.

2. Implementation of Wavelet Analysis using Filters

Equations (2.21) and (2.22) indicate that the scaling and wavelet coefficients at scale j are obtained by convolving the expansion coefficients at scale $j+1$ with the recursion coefficients $h(n)$ and $w(n)$[4]. Mallat [6] first implemented a wavelet decomposition structure consisting of a scaling filter (lowpass) $h(n)$ and a wavelet filter (highpass) $w(n)$. These two filters are related as given by

$$\begin{cases} w(n) = \pm(-1)^n h(N-n) \\ \sum_n h(n) w(n-2k) = 0 \\ \sum_n w(n) = 0 \end{cases} \tag{2.23}$$

where N is the length of the filter. The above equation shows that the high pass filter is a *mirror filter* of the low pass filter. The filter pairs are referred to as the quadrature mirror filter (QMF). The decomposition structure is shown in Figure 2-6.

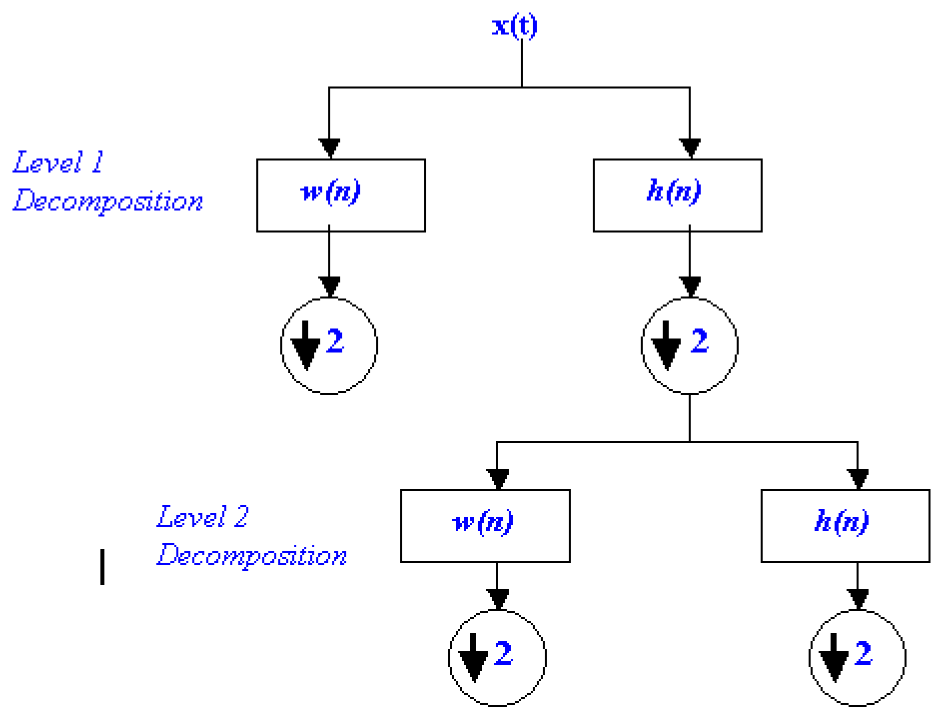


Figure 2-6. Mallat Wavelet Decomposition of Signal *x(t)* in Two Levels.

The above procedure is followed in reverse order for signal reconstruction. The reconstruction process in this case is made easy due to the fact that the filters form orthonormal bases. The signals at every level are up-sampled by two, passed through the synthesis filters $w^{'}(n)$ and $h^{'}(n)$ and then added.

Figure 2-7 shows the scaling and wavelet function of Daubechies'3 (db3) and their lowpass and highpass filters for signal decomposition or reconstruction. Note that the filter coefficients are time reversed between the decomposition and reconstruction operations.

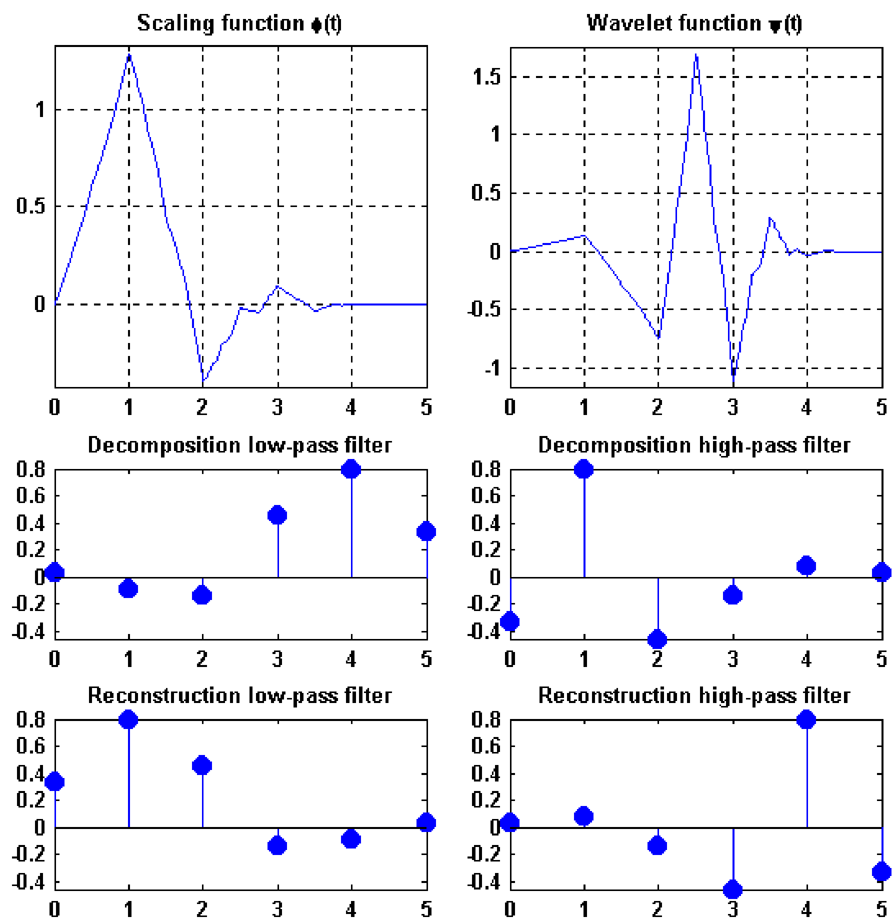
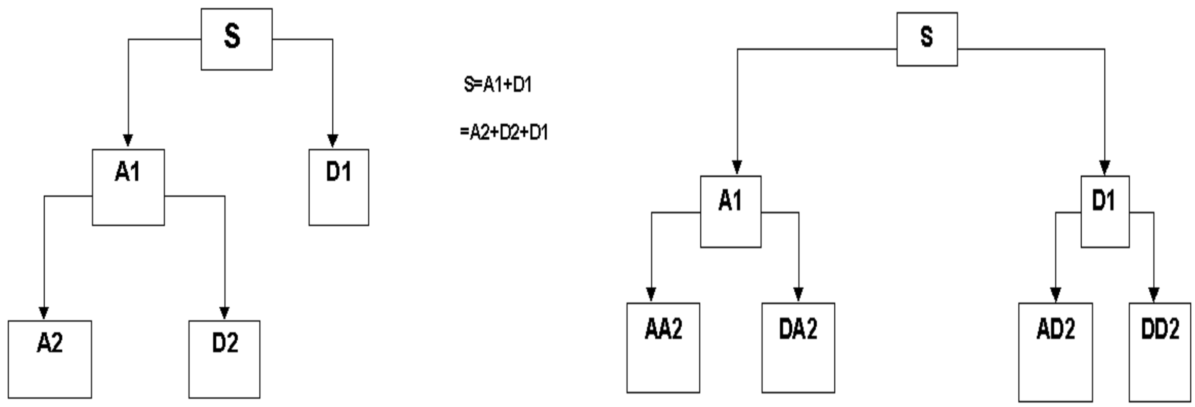


Figure 2-7. Scaling and Wavelet Function of db3 with their Decomposition and Reconstruction Filters.

Another way to decompose a signal using wavelets is the *wavelet packet* decomposition. In this case, the approximation and the details are further symmetrically decomposed into approximation and details. This technique gives more flexibility for signal representation than Mallat's decomposition. In Mallat's n-level decomposition, there are n+1 possible ways to decompose a signal. With n-level wavelet packet decomposition, there are more than $2^{2^{n-1}}$ different ways to decompose the signal. Figure 2-8 illustrates the difference between the two techniques. For instance, the signal, in the case of packet analysis, can be represented as: A1+AD2+DD2 or A1+D1 or AA2+DA2+D1.



(a) Mallat 2-Level Decomposition (b) Wavelet Packet 2-Level Decomposition

Figure 2-8. Comparison between Mallat Wavelet Decomposition and Wavelet Packet Decomposition.

E. IMAGE PROCESSING USING WAVELET ANALYSIS

Wavelet decomposition can be extended to two-dimensional signals, such as images. Like in the case of one-dimensional processing, analysis and synthesis using Mallat's or wavelet packets method can be implemented.

In practice, there are two ways to realize the subband decomposition of an image. The first is to use two-dimensional wavelet filters and the second is to separately transform the rows and columns with one-dimensional filters. A decomposition based on the latter approach was proposed by Mallat [6]. As shown in Figure 2-9, rows and columns are filtered using one-dimensional quadrature mirror filters *w* and *h.*

The *LL*, *LH*, *HL*, *HH* sub-images of Figure 2-9 are obtained by lowpass filtering of rows and columns, lowpass filtering of rows and highpass filtering of columns, highpass filtering of rows and lowpass filtering of columns, and highpass filtering of rows and columns, respectively. In practice, the *LL* sub-image gives the approximation of the image (low frequencies), *LH* the horizontal details, *HL* the vertical details and *HH* the diagonal details (high frequencies). The above decomposition is sometimes represented as shown in Figure 2-10. The approximation sub-image (*LL*) obtained in this fashion can be further filtered and subsampled to obtain four more sub-images. This process can be continued until the desired subband structure is obtained. In order to implement wavelet packet decomposition of an image, we have to follow the above process for all the sub-images of Figure 2-10.

Following the above procedure in reverse order, the image can be reconstructed from the decomposed components or sub-images. The synthesis filters, like in the case of one-dimensional processing, are identical to the analysis filters except for a time reversal.

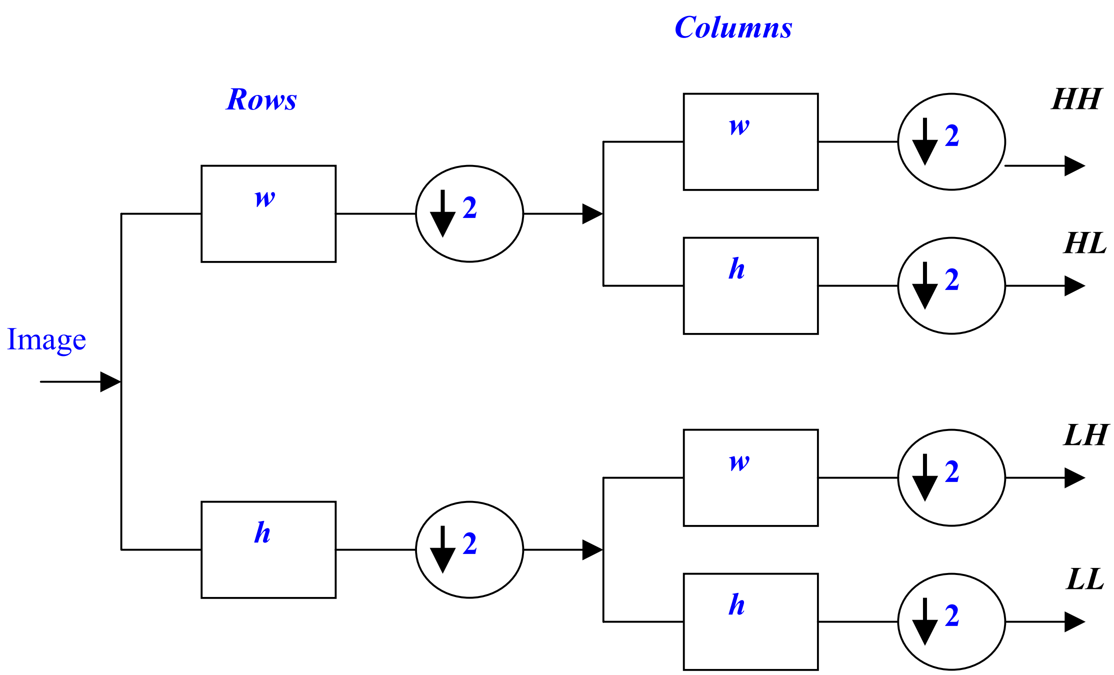


Figure 2-9. Decomposition of an Image using Mallat Decomposition.

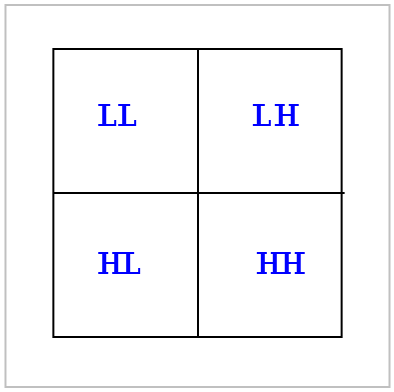


Figure 2-10. First Level Decomposition

Figure 2-11 shows the wavelet representation of an image decomposed on two resolution levels using Mallat's method. The upper left image of the figure is the original. The lower right one is the decomposition of the original image, and the lower left is the synthesized image. We observe that the synthesized image looks the same as the original and preserves almost all the details. The pattern of arrangement of the sub-images is as shown in Figure 2-10. The *LL* sub-image (approximation) of the original at the first resolution level is further decomposed into approximation and detail sub-images. The approximation at the second resolution level is shown as the upper right image of the figure.

Figure 2-12 shows the decomposition of the same image using wavelet packets on two resolution levels. As the decomposition tree at the upper left corner of the figure shows, after the initial decomposition each subband is further decomposed to four subbands. The lower left image shows the packet (2,0), which is the approximation sub-image of the LL subband of the first level decomposition.

The discrete wavelet transform (DWT) can be used to reduce the image size (image compression) without losing significant image quality. For a given image, the DWT can be computed, and all values of the DWT that are below a certain threshold can be discarded. Only those DWT coefficients that are above the threshold are saved, and during the image reconstruction process, each row and column is first padded with as many zeros as the number of discarded coefficients and then the inverse DWT is applied to reconstruct each row and column of the original image. Image compression using wavelet decomposition is the topic of the next chapter in which the JPEG2000 still image compression standard is described.

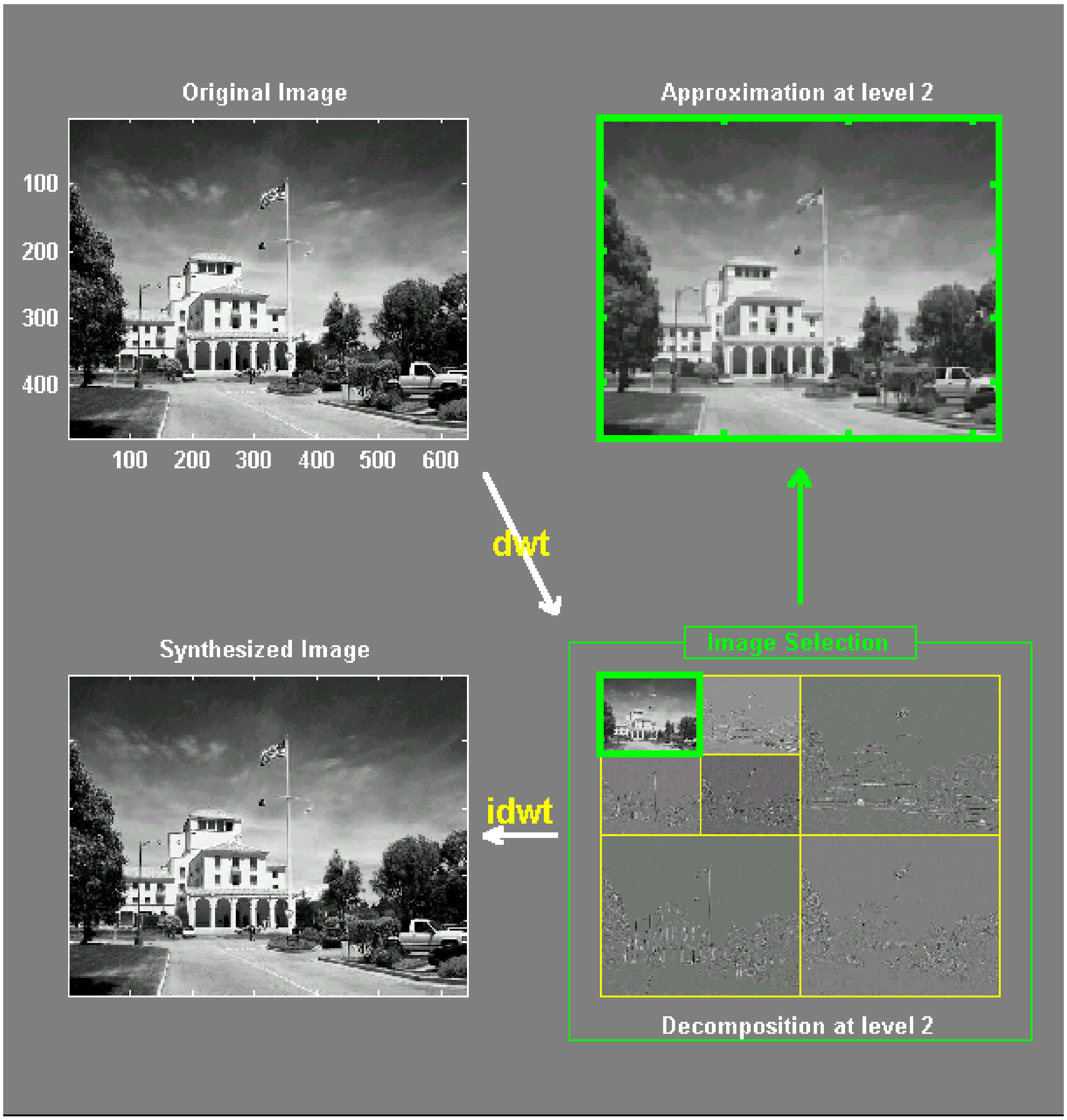


Figure 2-11. Two-Dimensional Subband Decomposition using Mallat's Method.

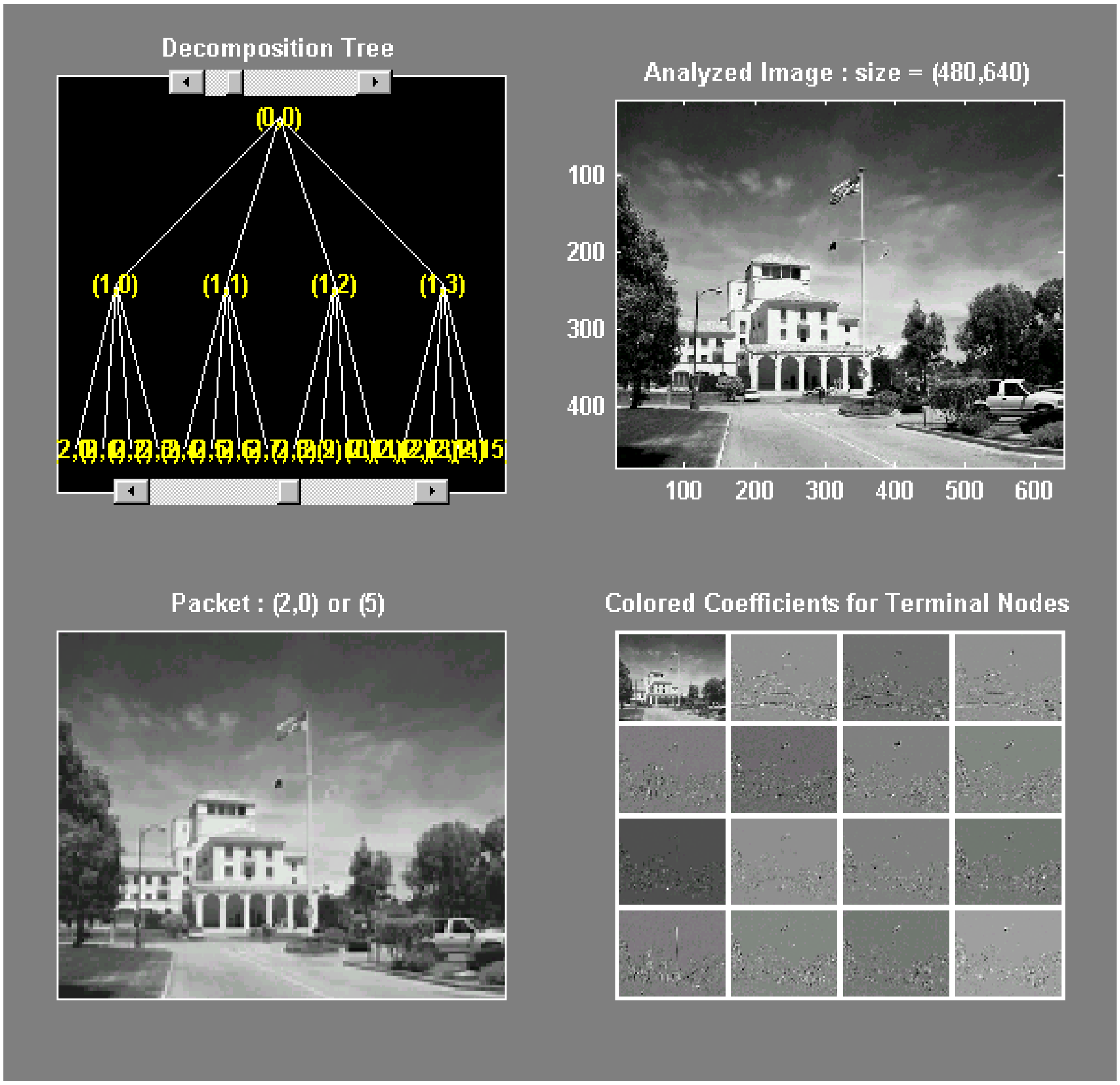


Figure 2-12. Two-Dimensional Subband Decomposition using Wavelets Packets.

F. SUMMARY

This chapter provided the basic concepts of wavelet analysis. The purpose was to highlight the main ideas and introduce the terminology. Wavelets offer a powerful tool for signal and image processing. Wavelets provide more accurate time and frequency representation than other signal analysis methods and can handle signals with discontinuities. The next chapter briefly describes the JPEG2000 still image compression standard, which is based upon wavelet decomposition and synthesis of images.

THIS PAGE INTENTIONALLY LEFT BLANK

III. THE JPEG2000 STILL IMAGE COMPRESSION STANDARD

A. INTRODUCTION

With the increased use of multimedia technologies, image compression requires greater performance as well as new features. In order to address this need in the specific area of still image compression, a new standard called *JPEG2000* is currently being developed. The new standard is intended to complement the existing DCT-based JPEG standard [7].

JPEG2000 is suitable for different types of still images, such as bi-level, gray-level and multi-component. It supports natural images, scientific, medical and text, and allows different imaging models, such as image library, and real-time transmission through channels with limited bandwidth [1]. JPEG2000 provides low bit-rate operation with rate-distortion and image quality performance superior to the existing JPEG standard. Some of the features of JPEG2000 are [7]:

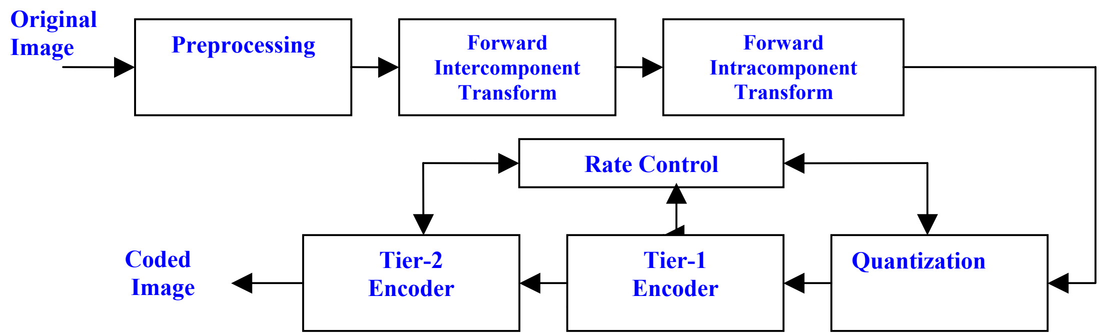
- State-of-art low bit rate compression performance
- Progressive transmission by quality or resolution
- Lossy and lossless compression
- Random access to bitstream
- Pan and zoom (while the compressed data is not entirely decompressed)
- Region of interest (*ROI*) coding by progression

This chapter introduces the JPEG2000 standard. It also presents the error resilient tools to be used in Chapter IV for the simulation of image transmission through unreliable networks. The performance of JPEG2000 (using *VM8.5 Part II* source code) is compared with the widely used *JPEG Baseline* source code.

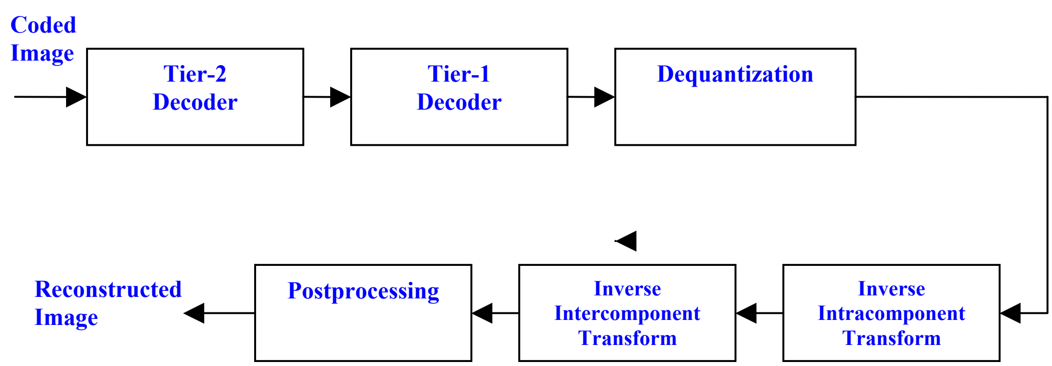
B. STRUCTURE OF THE STANDARD

1. JPEG2000 Codec

JPEG2000 is based on wavelet/subband coding techniques. The schematic diagrams of the encoder and the decoder are shown in Figure 3-1. In the following subsections, the functionality of each of the blocks in Figure 3-1 will be described. The discussion here focuses on the encoder since the decoder simply undoes the encoding process of the image. Parts of the decoder that work differently are mentioned and briefly described.



(a)



(b)

Figure 3-1. JPEG2000 Codec Structure. The Structure of the (a) Encoder and (b) Decoder from [8].

a. Preprocessing

An image typically consists of one or more components; for example, a RGB image has three components and a grayscale image has only one component. Components are allowed to have a different number of bits per component sample (1 to 32 bits/sample). Since different components may have different sizes, JPEG2000 provides a common description using a system called the *canvas coordinate system*. Figure 3-2 illustrates the canvas coordinate system as well as the position of the image component on the canvas. The origin of the canvas is the upper left hand corner of the figure, and the lower and right hand boundaries are defined by the image [7].

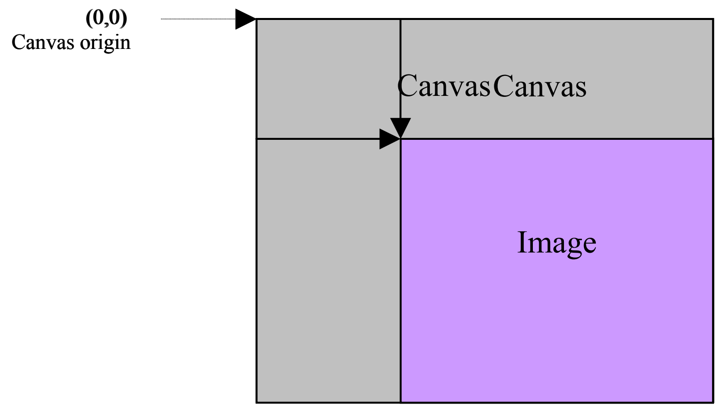


Figure 3-2. Placement of an Image on the Canvas from [7].

After its placement on the canvas, the image component is divided into nonoverlapping segments called tiles, which are coded independently (see Figure 3-3). All the tiles need to be of the same size. Figure 3-3 shows that in case the tiles in the image boundaries cannot be all equal, then the image is zero padded to ensure that all tiles are of the same size. Tiling requires low memory and provides the ability for spatial random access [9].

In the description of the following blocks of Figure 3-1, we will present the details of processing one tile of the image since the same applies to all tiles.

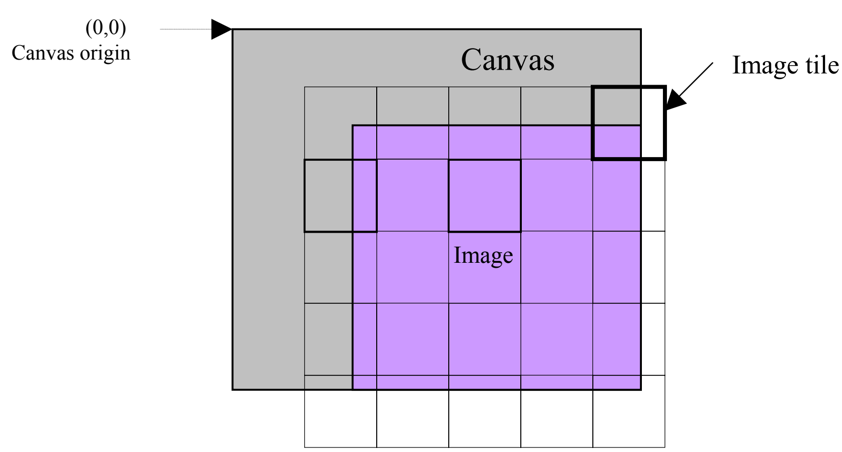


Figure 3-3. Tiling of the Images in JPEG2000 [7].

b. *Intercomponent Transform*

Intercomponent transform is applied to multicomponent images, such as RGB, in order to decorrelate their components. JPEG2000 allows two types of component transforms: *YCrCb* transform and *reversible component transform.* The YCrCb transform is similar to the one used in JPEG, and the reversible component transform allows both lossy and lossless reconstruction [7], [9], [10], [8].

c. *Intracomponent Transform*

In the intracomponent transform, the image component values are subjected to wavelet decomposition. Using wavelet filters, components within the tile are mapped into the wavelet domain. Presently, two kinds of wavelet filters are used in JPEG2000. The default is the Daubechies 9-tap (lowpass)/7-tap (highpass) filter, which implements a nonreversible floating point wavelet transform. The other wavelet filter is the Daubechies 5-tap/3-tap filter, which implements a reversible (integer-to-integer) wavelet transform [7]. Tables 3-1 and 3-2 list the coefficient values of Daubechies 9/7 and Daubechies 5/3 filters, respectively.

n	$h[n]$	$w[n]$
0	0.6029490182363576	1.115087052456994
±1	0.2668641184428723	-0.5912717631142470
±2	-0.07822326652898785	-0.05754352622849957
±3	-0.01686411844287495	0.09127176311424948
±4	0.02674875741080976[5]	

Table 3-1. Daubechies 9/7 Analysis Filter Coefficients for Reversible Wavelet Transform of JPEG2000 from [7].

n	$h[n]$	$w[n]$
0	6/8	1
±1	2/8	-1/2
±2	-1/8	0

Table 3-2. 5/3 Analysis Filter Coefficients for Reversible Wavelet Transform of JPEG2000 from [7].

After transforming the image tile components to the wavelet domain, each subband of every resolution level is further partitioned into blocks called *precincts* [10]. The size of all precincts must be the same and need to be a power of 2. Figure 3-4 shows a precinct partition for a single resolution level of an image tile. Compressed data of precincts will later form a *packet*. The above partitioning plays an important role in organizing the data within a code-stream. Precincts are further partitioned into *code-blocks*, which form the smallest geometric structure of JPEG2000. The main advantage of code-blocks is that they provide fine grain random access to spatial regions and also help the *quantization* process and *bit-plane coding*, which are described in the following subsections [10].

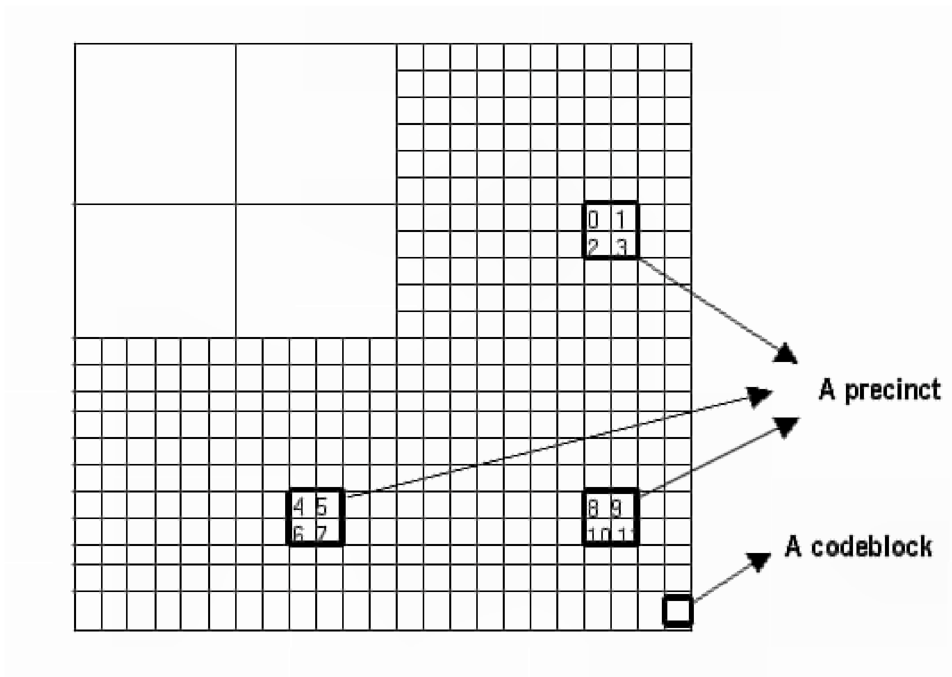


Figure 3-4. Partitioning of a Wavelet Subband into Precincts and Code-Blocks from [10].

d. Quantization/Dequantization

In the encoder, after all partitions are formed, the resulting coefficients are quantized. There are two methods of coefficient quantization: scalar dead-zone quantization and trellis coded quantization. A different quantizer step size is applied for the coefficients of each subband. Both methods quantize wavelet coefficients $x_i[n]$ to form sequences of indices $q_i[n]$ for each code-block B_i. Since scalar dead-zone quantization is the default quantizer in the encoder as well as the one used for simulations in this thesis, we will further examine it here [7].

The scalar dead-zone quantizer relates the sample values, $x_i[n]$, to indices, $q_i[n]$, as follows [7], [8]:

$$q_i[n] = \operatorname{sgn}(x_i[n]) \cdot \left\lfloor \frac{|x_i[n]|}{\Delta_b} \right\rfloor \tag{3.1}$$

where Δ_b is the scalar quantizer step size for the subband that contains block B_i. In the decoder, a coefficient is reconstructed from the corresponding index using the expression [7], [8]

$$\hat{x}_i[n] = (q_i[n] + r \cdot \operatorname{sgn} q_i[n]) \cdot \Delta_b \tag{3.2}$$

where r is the bias parameter, which is typically equal to ½.

e. Tier-1 Coding

The Tier-1 coding process is a *bit-plane coding* technique. It is based on the Embedded Block Code with Optimize Truncation (EBCOT) algorithm and is performed independently on each code-block. First, each code-block is scanned as shown in Figure 3-5. Then, bit-planes for each code-block are created. Bit-planes are defined as a sequence of arrays; each array contains one bit of each quantized index. The first of the arrays contains the most significant bit of all the indices, the second contains the next most-significant bit and the last contains the least significant bits. The number of bit-planes will be transmitted as side information [11].

In order for these bit-planes to be encoded, there are three passes per bit plane, starting with the most significant bit-plane. The three passes are the s*ignificant pass*, the *refinement pass* and the *cleanup pass* [1], [7], [12].

After the bit-plane coding is completed, all the resulting symbols are entropy coded with an adaptive binary arithmetic coder. An option to bypass arithmetic coding for some of the least significant bit planes exists (referred to as Lazy Mode) [13], [8].

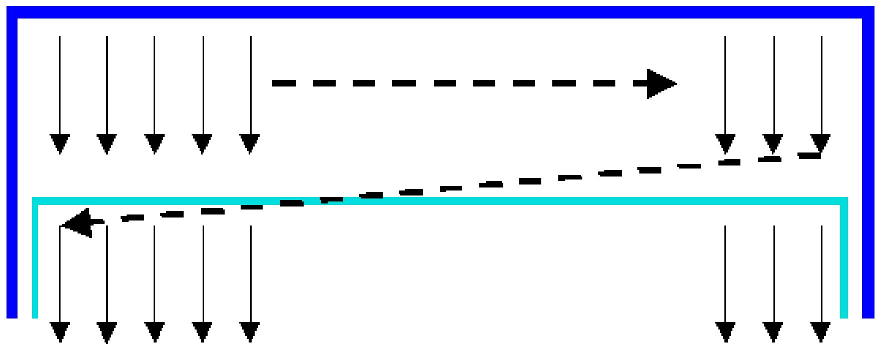


Figure 3-5. Scan Pattern Tier-1 Coding inside a Code-Block from [7].

f. Tier-2 Coding

Tier-2 receives the embedded bitstream of each code-block from Tier-1, composes a collection of N quality layers Q_i, and truncates each layer at a suitable truncation point depending on the desirable bit rate. Figure 3-6 illustrates the quality layers of code-blocks as rows of blocks and the truncation of some layers as shaded area. The layers are formed in a way such that layer Q_1 represents the most important data of each code-block while Q_N represents the finest details. During decoding, the reconstructed image quality improves with each successive layer reception [13].

A sequence of twelve coded code-blocks of the same layer, the same resolution level, and specific precincts form a *packet*. An example of a packet is the coded data of the twelve code-blocks of Figure 3-4. The data inside a packet is ordered such that the contribution from the *LH*, *HL* and *HH* subbands appears in that order. Only those code-blocks that contain samples from the relevant subband, confined to the precinct, have any representation in the packet [7], [10].

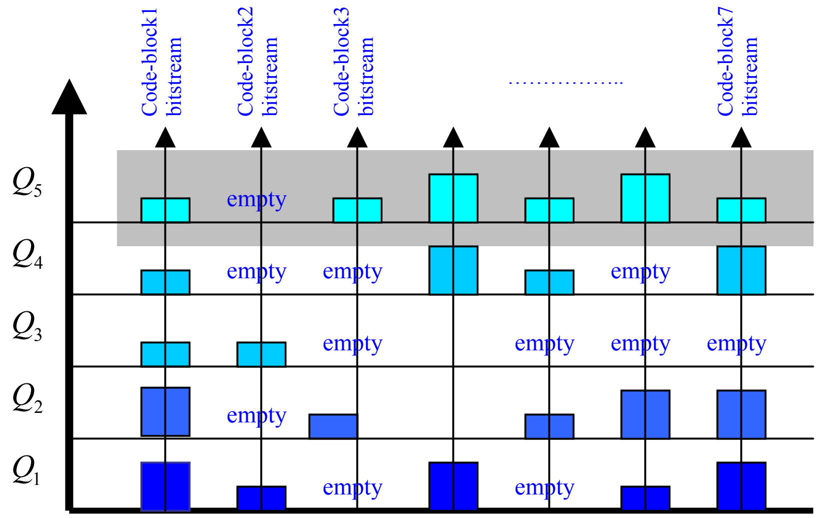


Figure 3-6. Progressive Embedded Code-Block Bitstreams in Quality Layers, from [13]. The Shaded Region indicates Discarded Blocks.

Figure 3-7 illustrates the above process in an image tile. The tile has been decomposed into three resolution levels. Each level contains four subbands (blue color represents LH subband, green represents the HL and red represents the HH subband) and each subband then contains a number of precincts (numbered blocks) whose sizes are equal to the approximation of the third level decomposition. Each packet will contain specific precincts of each subband of the particular resolution level. The first packet contains precinct 1, the second contains precincts 2, 3 and 4, the third contains precincts 5, 6 and 7, and so on.

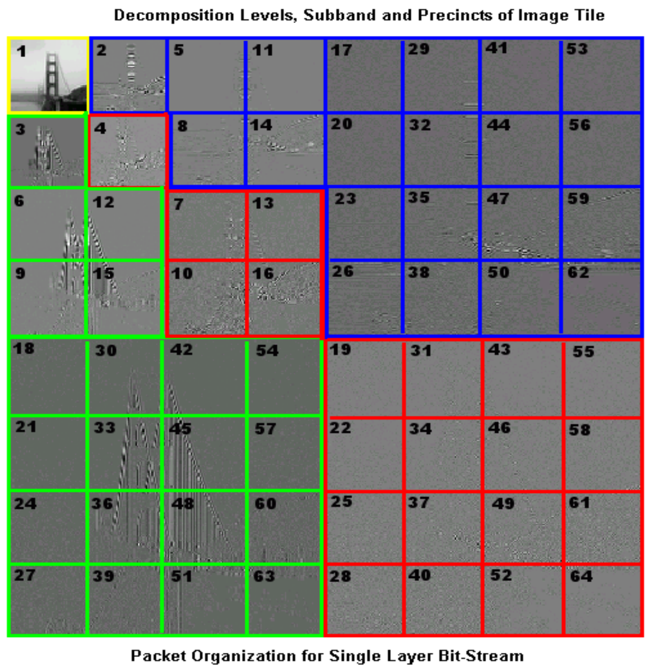


Figure 3-7. Tile of an Image Decomposed into Multiple Resolution Levels and Subbands, and Partitioned into Precincts. Numbers Indicate a Suggested Order for Bitstream Organization.

2. Coded Bitstream Organization

Figure 3-8 illustrates the basic organization of a JPEG2000 bitstream produced by the encoder. The bitstream consists of a global header corresponding to the whole image, followed by one or more sections depending of the number of tiles of the original image. Each such section consists of two parts. The first part consists of a start of tile marker, a tile header and the start of sequence marker. The second part includes the layered representation of the code-blocks belonging to that tile, which is organized into packets as previously described [7]. The optional resynchronization (resync) markers indicated in Figure 3-8 will be explained later in the following section on the error resilient capabilities of JPEG2000.

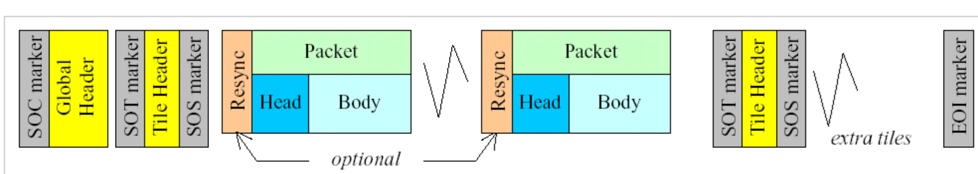


Figure 3-8. Basic Organization of JPEG2000 Bitstream from [14].

The structure of the bitstream received by the decoder is based on the packets and their organization in layers. The received image may be a single layer bitstream organization (Figure 3-8), a multi-layer "*resolution progressive*" bitstream organization, or a multi-layer "*SNR progressive*" bitstream organization [14]. Appendix A provides examples of resolution and SNR progressive image decoding.

C. ERROR-RESILIENT TOOLS

When transmitted over unreliable wired or wireless channels, packet losses may occur in an image bitstream. Existing transmission protocols for networks suffer from packet losses due to network congestion. Likewise, the wireless networks are subject to fading, interference or burst errors because of multipath propagation. Channel coding techniques can be applied to reduce the bit error rates; however, the residual bit error rate may have a significant impact on image quality [15].

Many coding techniques are not robust to errors by nature. For example, predictive coding and variable length coding are able to provide high compression but are not resilient to errors, such as packet loss [15]. On the other hand, JPEG2000 incorporates error resilience at the source coding level in order to overcome the problem of burst errors and packet losses.

1. JPEG2000 and Error Resilient Tools

JPEG2000 provides a variety of error resilient tools. These tools can be classified into three major types: resynchronization for packet protection, segmentation for code-block protection, and error resilient termination for code-block protection. In principle, these tools detect and locate errors, support fast resynchronization and limit the loss of information [15], [12], [7].

a. Resynchronization

Resynchronization tools attempt to establish the resynchronization between the decoder and the bitstream. They localize the error and prevent it from impacting the entire bitstream. An effective resynchronization mechanism makes error recovery and concealment easier [15].

As illustrated in Figure 3-8, resynchronization markers (*resync*) are optional and if used they have to be inserted before every packet in the bitstream. They consist of three bytes that define the correct order of the packets of each tile. When multiple tiles are present, the resynchronization marker sequence index is reset after the end of each tile. There is a flag in the global header of the bitstream, which indicates the use of the *resync* marker mechanism. In order to allow error resilient mechanism to locate the boundaries of each packet in case of error injection throughout the bitstream, a "byte-stuffing" is applied to the head and body bytes of each packet [14].

It was also mentioned earlier that the head and body of a packet have different sensitivity to error. For example, suppose there is a single bit error somewhere in the packet. If this occurs in the packet body, then only one code-block is affected. Since all code-blocks are coded independently and contain data that refers to a subband in a particular resolution level of a small part of a tile, the corruption is limited spatially and the result is a very small portion of the received image does not include some frequencies. On the other hand, the packet head contains information about the truncation points for every code-block, in every subband. In case that the head of the packet is corrupted with error injection, then all information in the current packet and all future packets from the same subband and resolution level are useless [14].

b. Segment Markers

Another error resilience mechanism at the code-block level is the segment marker. The use of segment markers is also optional. It consists of four-symbols that have to be inserted at the end of the normalization-coding pass of the tier-1 coding level. Its functionality is based on the correct decoding of the fixed pattern, "1010". In case that this pattern is not found, error detection is assumed, and the current coding pass as well as all the following coding passes will are discarded. Additionally if the segment markers is the only error resilient mechanism in use, then the two previous coding passes will also be discarded [14], [12].

c. Error Resilient Termination

JPEG2000's error resilient mechanism is based on the use of predictable truncation points of code-block layers. Generally, the encoder is free to terminate the code-block in any manner in order to achieve the desired bit-rate. The VM8.5 algorithm provides the ability for the encoder to use a predictable termination policy with which the decoder is familiar. The use of a specific termination pattern comes with a flag in the global header, which helps the decoder to take advantage of this and detect the errors.

Although not optimal, this termination policy has been selected because of its simplicity [7], [14]. The result of this error resilient termination is that in case an error is injected into the transmitted data then the decoder is always able to decode the received code-stream. The resulting image may have one or more tiles of lower quality, but it is always complete without any blank areas.

d. Example of Error Resilience in JPEG2000

In order to examine the performance of the error resilient tools in JPEG2000, a grayscale image is compressed to 2 bpp and transmitted through a simulated channel. Prior to transmission, the JPEG2000 bitstream is encoded using a rate ½ convolutional coder with a constraint length of 7, for additional redundancy. The simulation is repeated for the same image, through the same channel, but now using the error resilient tools of JPEG2000. The average residual bit-error-rate in both cases is 3×10^{-5}. Figure 3-9 shows the received images in both cases. The peak-signal-to-noise-ratio (pSNR) without error resilient tools is 28.60 dB while it is 34.70 dB with error resilient tools.

(a) Without Using Error Resilient Tools

(b) With Error Resilient Tools

Figure 3-9. Transmission of JPEG2000 Compressed Images through a Noisy Channel.

2. JPEG Error Resilient Tools

JPEG uses a block based discrete cosine transform and a variable length coder (Huffman or arithmetic). The resynchronization markers for JPEG are placed at the boundaries of every n^{th} block, where n is chosen at encoding time. When an error occurs, all information being decoded is discarded until the next valid start marker is reached. As a result, some blocks may be lost as shown in Figure 3-10 (a). Unlike JPEG2000 where a global view of the image is always obtained, the JPEG decoder may stop decoding and generate an empty strip [15]. Figure 3-10(a) shows a received image, coded without using restart markers. The image cannot be decoded entirely due to the loss of synchronization between blocks after error injections inside the bitstream. Figure 3-10(b) shows the same JPEG image after transmission through the same simulated channel but now enhanced with resynchronization markers every block. The image has some strips that cannot be correctly decoded, but the overall result is superior to the previous case.

(a) JPEG without Restart Markers

(b) JPEG with Restart Markers

Figure 3-10. Error Resilient Capabilities of JPEG Baseline Image Compression Standard.

D. COMPRESSION PERFORMANCE

1. JPEG2000 Compression Performance

In order to examine the performance of the JPEG2000 still image compression standard, a variety of grayscale (8 bpp) images were used as listed in Table 3-3. All the images are compressed with the JPEG2000 VM8.5 source code, and the peak-signal-to-noise-ratio (pSNR) in decibels (dB) between the compressed and the original was computed in MATLAB according to the following equation [16]

$$pSNR = 10\log_{10}\frac{\psi_{\max}^2}{\sigma_e^2}\ , \tag{3.1}$$

where $\psi_{\max}$ is the maximum intensity value of the image and σ_e is the mean squared error between two $M \times N$ images ψ_1 and ψ_2 as given by [16]

$$\sigma_e^2 = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}\left(\psi_{1_{ij}} - \psi_{2_{ij}}\right)^2 \tag{3.2}$$

Image	*Woman*	*Building*	*Leaves*	*Airplane*	*Boats*	*Tall ship*
Size	**5.52 MB**	**1.83 MB**	**759 KB**	**707 KB**	**726 KB**	**388 KB**

Table 3-3. Sizes of the Original Images used for JPEG2000 Performance Evaluation.

Figure 3-11 shows the results of the images compressed in the range of [0.025, 2] bpp. We observe that as the bit-rate of the compressed image decreases, its pSNR also decreases since the truncation of more layers of the code-block of the bitstream is necessary. It is important to note that compression at bit resolutions less than 0.15 bpp using JPEG is difficult and the resulted pSNRs are very low, whereas with JPEG2000, the quality is reasonable (by visual evaluation) even at a bit resolution of 0.025 bpp.

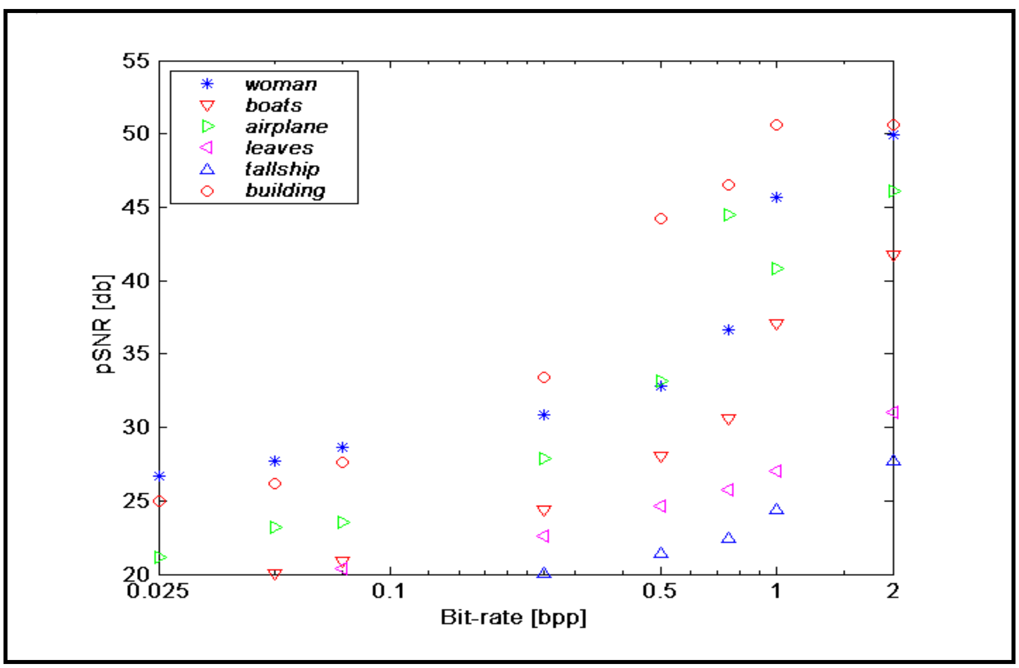


Figure 3-11. JPEG2000 Compression Performance for Different Images.

Additionally the reversible wavelet transform with integer-to-integer kernels (*W5x3*) was tested and the file sizes with maximum achievable lossless compression are listed in Table 3-4. For the sizes of the original images, see Table 3-3.

Image	*Woman*	*Leaves*	*Building*	*Tall Ship*	*Airplane*	*Boats*
Size	1.7084 MB	1.1172 MB	159.29 KB	395.43 KB	430.06 KB	124.83 KB
Compression ratio	3.23:1	1.63:1	4.76:1	1.78:1	1.69:1	3.1:1

Table 3-4. File Sizes of Compressed Images Using the Lossless Mode of JPEG2000.

2. Comparison between JPEG2000 and JPEG

We now compare the compression performance of JPEG2000 with that of JPEG. For JPEG compression, the *JPEG V6* baseline mode is used. For JPEG2000, progressive mode has been optimized for 0.0625, 0.125, 0.25, 0.5, 0.75, 1.0, 2.0 bpp by using 9-tap/7-tap filters.

Figures 3-12 and 3-13 show the performance comparison for the two standards in terms of pSNR measurements for different levels of compression. The results obtained for the images indicate that JPEG2000 provides better compression performance than JPEG. For bit-rates up to 0.5bpp, compression with very low pSNRs resulted for both standards, even though JPEG2000 compressed the images with less distortion. For compression ratios above this value, the JPEG standard produced poor results while JPEG2000 maintained the quality at a high level. With the JPEG image compression standard, the user is not able to define a desirable compression ratio. The only variable that JPEG accepts is the expected quality of the compressed image as a value between 100 (no compression) and 0 (worst quality). Due to this inflexibility, JPEG is not able to compress the image down to more than 0.1 bpp. For this compression level, the image quality is poor because of the blocking artifacts. Figures 3-14 and 3-15 show compressed images at the maximum affordable compression for each standard. Appendix A provides additional results for some other images.

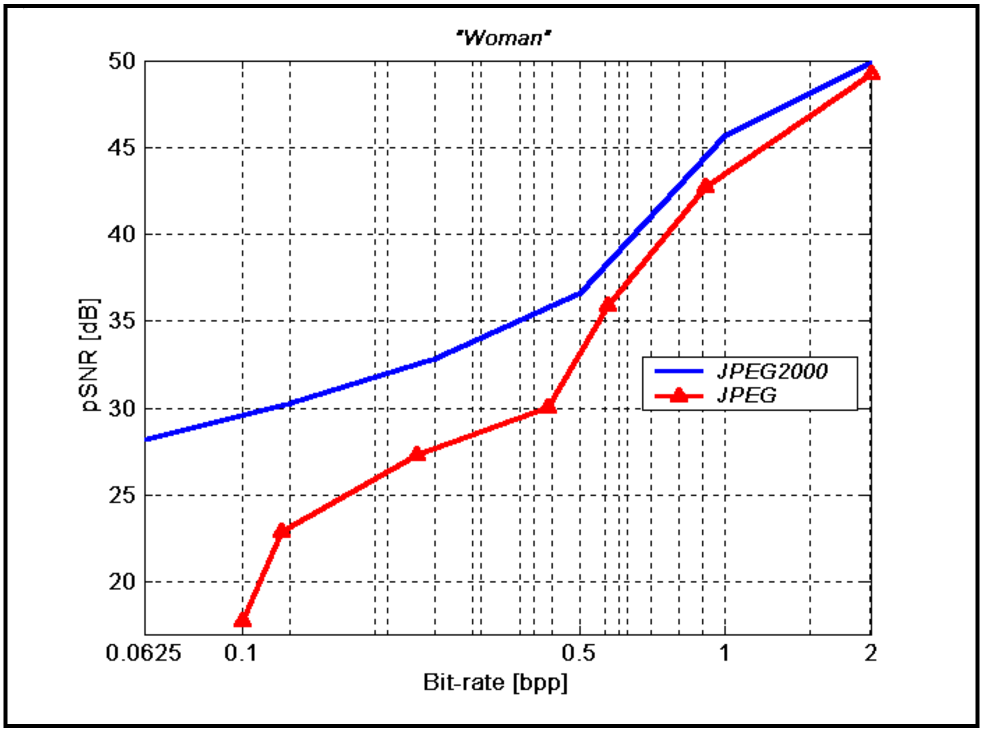


Figure 3-12. Rate-Distortion Performance for JPEG and JPEG2000 on Grayscale "Woman" Image.

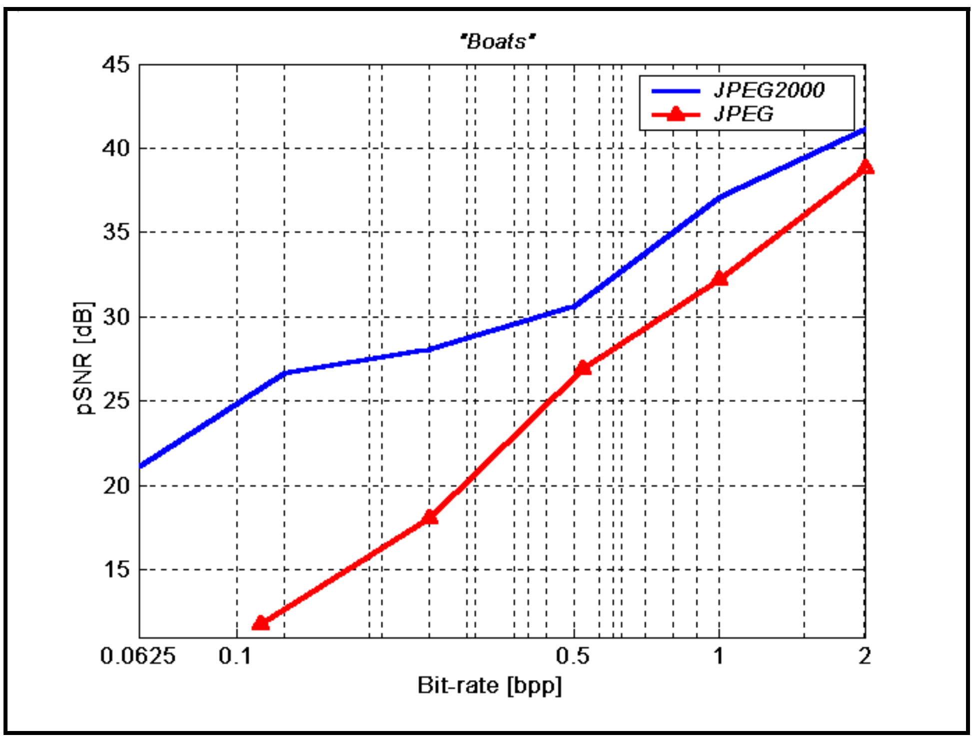


Figure 3-13. Rate-Distortion Performance for JPEG and JPEG2000 on Grayscale "Boat" Image.

(a) JPEG Image (b) JPEG2000 Image

Figure 3-14. Compression ratios of $71:1$ for JPEG and $320:1$ for JPEG2000 for Image "Boats".

(a) JPEG Image (b) JPEG2000 Image

Figure 3-15. Compression ratios of $80:1$ for JPEG and $320:1$ for JPEG2000 for Image "Woman".

The above performance advantage of JPEG2000 comes at the expense of memory, execution time and complexity (encoding and decoding) [17]. Memory usage as well as execution time is independent of the bit-rate for the encoder. The memory usage and the execution time are increased by a factor of 40 and 34, respectively, for the encoder of JPEG2000 relative to JPEG. The results for the decoder are better but as the bit-rate increases, the memory usage and the execution time increase. However, the increase in execution time for the decoder is no greater than a factor of 8 [1]. Table 3-5 provides measured execution times of the JPEG2000 encoder compared to JPEG for three different sizes of the image "*Woman*" and for bit resolutions of 0.5 bpp.

Image	**Dimensions**	**Size**	**JPEG Encoder Time**	**JPEG2000 Encoder Time**
Woman	2032×2800	5.42MB	0.2 sec	3.3 sec
Woman	4064×5600	21.70MB	2 sec	16.5 sec
Woman	6096×8400	48.83MB	2.5 sec	32.5 sec

Table 3-5. Execution Time for JPEG2000 Versus JPEG Encoder for Different Size Images (Platform Pentium III, Processor 750 MHz).

3. Region of Interest (ROI), a Unique Capability of JPEG2000

One interesting and unique feature of JPEG2000 is its capability to define regions of interest and code the selected regions at a better quality than the rest of the image. The regions can have any shape and size. This technique is employed in the JPEG2000 coder by defining the coefficients of the region of interest as more important than the rest prior to Tier-1 coding. In order to accomplish this, it scales all the coefficients of the ROI upward by a power of two and leaves the rest of the coefficients the same. During the embedded coding process, those coefficient bits are placed in the bitstream before the background parts of the image. Thus, the ROI is decoded before the rest of the image. Regardless of scaling, a full decoding of the bitstream results in reconstruction of the whole picture with the highest fidelity available. If the bitstream is truncated or the encoding process is terminated before the whole image is fully encoded, the ROI will have a higher fidelity than the rest of the image [18]. Figure 3-16 shows an example of ROI coding with the JPEG2000 VM8.5 coder. The whole image is compressed to an average bit allocation equal to 0.25 bpp. The region of interest here is square and clearly the quality of the image within this region is superior to the rest of the image outside of it. Appendix A includes more examples of ROI coding for regions of different shapes.

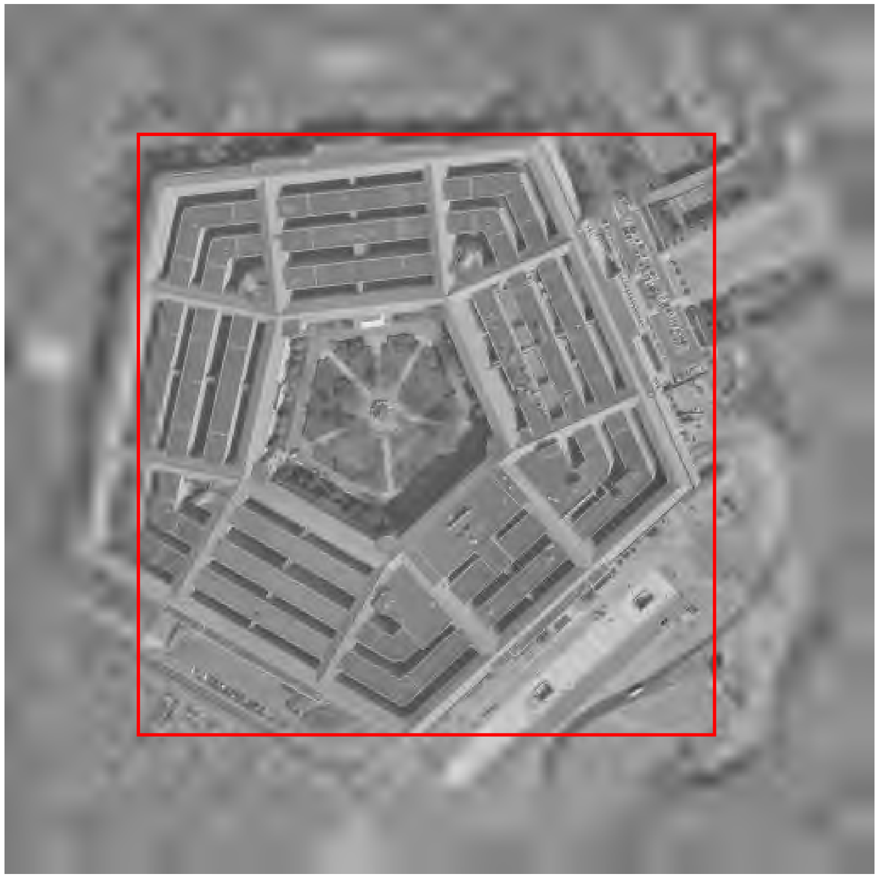


Figure 3-16. Reconstructed Satellite Image of Pentagon in which a ROI of Rectangular Shape has been Defined in the VM8.5 Encoder.

E. SUMMARY

This chapter began with a high-level introduction to the JPEG2000 standard and proceeded to describe the JPEG2000 VM8.5 codec briefly. JPEG2000 is a new standard for still image compression suitable for such applications as the Internet, wireless transmission, digital library access and medical imaging. It supports both lossy and lossless compression and is progressive by resolution and quality. Comparative results exhibit that image compression of JPEG2000 is superior to that of the existing standard JPEG.

THIS PAGE INTENTIONALLY LEFT BLANK

IV. STILL IMAGE TRANSMISSION OVER NOISY CHANNELS

The need for robust transmission of images through wireless networks has arisen in recent years because of the tremendous growth in the area of mobile communications. In general, the wireless environment suffers from limited bandwidth resources (especially for military operations) and is characterized by high bit-error behavior. It is, therefore, imperative that some form of error control has to be used in order to achieve reliable transmission.

Many coding schemes have been developed during the last few years in order to protect the digital data from transmission errors. These coding schemes add redundant bits to the bitstream before the transmission, which will help enhance the receiver's ability to detect and possibly correct errors. Even though error control schemes enhance reliability of bit transmission, several trade-offs exist. A great amount of redundancy leads to low effective throughput in the channel. Additionally, many error control schemes introduce delay, which makes these schemes unattractive for some real-time applications [15].

In order to overcome the problem of channel errors and provide robust bitstream syntax, JPEG2000 introduces error control schemes at the source coding level. These schemes are optional (as described in Chapter III) and are able to provide error detection and error correction for image transmission through error-prone environments.

This chapter examines the forward error correction (FEC), automatic repeat request (ARQ), and hybrid automatic repeat request (HARQ) error control schemes. A wireless channel model based on a two-state Markov process is described. The error resilient mechanisms of JPEG2000 are then tested in combination with FEC.

A. FORWARD ERROR CORRECTION

Error control codes are used to format the transmitted information so as to decrease the effect of noise. This is accomplished by inserting *controlled redundancy* into the transmitted information stream that allows the receiver to detect and possibly correct errors. Many different types of error control codes are available. The most popular are:

linear block codes, cyclic codes, Golay and Reed-Muller codes, BCH and Reed-Solomon codes, convolutional codes and turbo codes [19].

An introduction to the convolutional coder follows as it is used later in this chapter for the simulation of image transmission through different kinds of wireless channels.

1. Convolutional Codes

A convolutional code is generated by passing the information sequence to be transmitted through a linear finite state shift register. If n is the number of output bits for a sequence of k input bits, the code rate is defined as $r = k/n$. The constraint length K of a convolutional encoder is the maximum number of bits in the output stream that can be affected by one input bit and is given by [19], [20]:

$$K \triangleq 1 + \max_i m_i \qquad (4.1)$$

where m is the number of shift registers of each branch. Figure 4-1 illustrates a rate ½ convolutional encoder with a constraint length of 4. The output sequence v has a length of 2 while the input sequence u is a single bit.

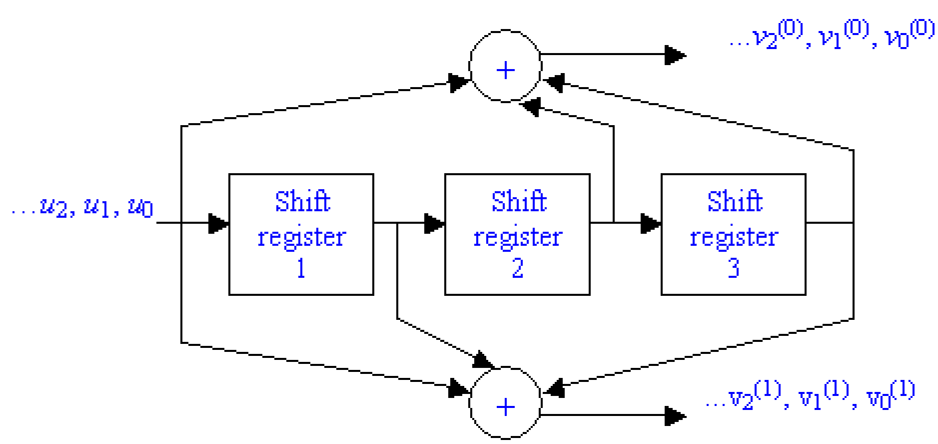


Figure 4-1. Rate ½ Convolutional Encoder with a Constraint Length of 4, from [19].

Another important parameter of convolutional codes is the free distance d_{free}, which is defined as the minimum Hamming distance between all pairs of the convolutional coded words [19] :

$$\begin{aligned} d_{free} &\triangleq \min\left\{d\left(v^{'}, v^{"}\right) \mid v^{'} \neq v^{"}\right\} \\ &= \min\left\{w\left(v\right) \mid v \neq 0\right\} \end{aligned} \tag{4.2}$$

Convolutional codes use the Viterbi algorithm as the maximum likelihood decoding algorithm [19], [20]. The error-correction capability of the Viterbi decoder increases as d_{free} increases. With the use of a convolutional encoder and Viterbi decoding algorithm, we can correct up to t errors occurring within a time span corresponding to one constraint length, where [19]

$$t \leq \left\lfloor \frac{d_{free} - 1}{2} \right\rfloor \tag{4.3}$$

a. Interleaving for Coded Systems

When multipath propagation effects characterize the channel, the received signal amplitude fluctuates over short time periods compared to the message duration, which results in burst errors. In order to decorrelate burst errors, interleavers are used.

An *interleaver* is a device that jumbles the symbols from several different codewords so that the symbols from each codeword are separated during transmission. A *deinteleaver* reverses the process before passing the symbols to the decoder. In this way, error bursts introduced by a channel are spread across a number of different codewords, hence the combination of interleaver/deinterleaver effectively converts a bursty channel into a random channel [20].

The interleaver can take one of two forms: a block structure or a convolutional structure. A *block interleaver* formats the encoded data in a rectangular array of m rows and n columns. The bits are read out column-wise and transmitted over the channel. At the receiver, the deinterleaver stores the data in the same rectangular array format, but it is read out row-wise [20]. As a result, a burst of errors of length $l = mb$ is broken up into m bursts of length b, which could be corrected depending on the error correction capability of the code according to Equation (4.3).

A t-error correcting code with $(m \times n)$ block interleaving can correct an error burst of length $l \leq mt$ bits. The length of the shortest error burst that exceeds the error correcting capability of the code and will cause at least one decoding error is $mt + 1$.

B. ARQ PROTOCOL

The simplest form of error control for a full duplex channel is the automatic-repeat-request (ARQ) protocol. The transmitted data frames are encoded for error detection and if an error is detected at the receiver, a retransmission of the frame is requested. There are three types of ARQ: the stop-and-wait, the go-back-N, and the selective-reject [19]. The stop-and-wait ARQ is used later in this chapter for simulations.

A common error detecting code used for the ARQ scheme is the *cyclic redundancy check* (CRC). From an *n*-bit block of data, the transmitter generates an *(m-n)* –bit frame check sequence and transmits an *m*-bit frame. The receiver uses an error detection scheme to determine if the frame is error free [21].

Figure 4-2 shows the time sequence diagram of the stop-and-wait protocol. The receiver replies with an acknowledgment when no errors are detected; if errors are detected in the received frame or the receiver does not receive the frame after a period of time, it returns a negative acknowledgment and requests retransmission of that frame. In case the transmitter does not receive any kind of acknowledgment within a predefined time, the transmitter sends the same frame again.

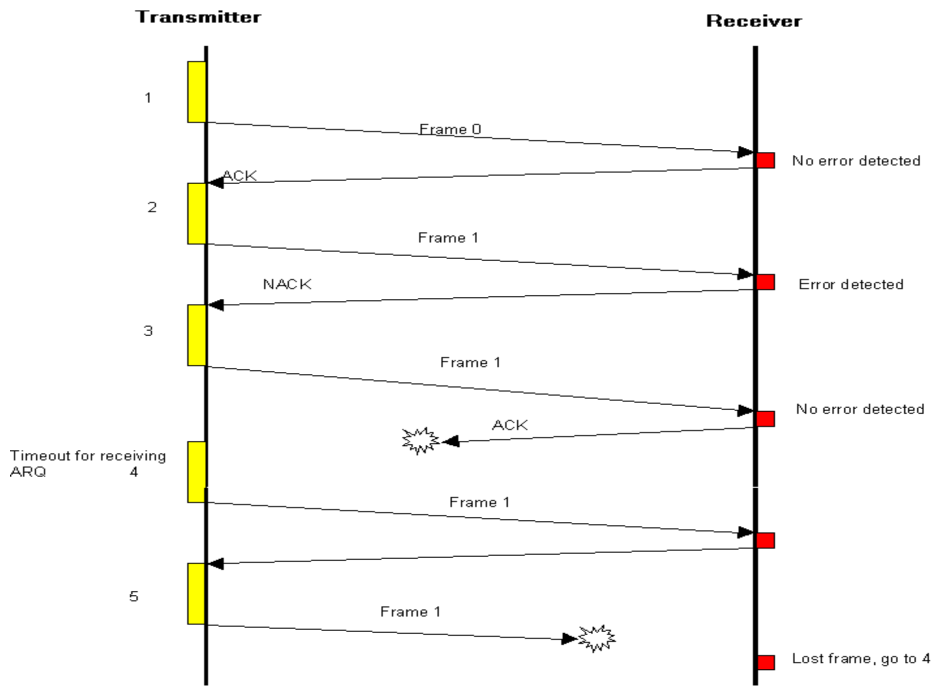


Figure 4-2. Time Sequence Diagram of Stop-and-Wait ARQ Protocol.

Assuming that the time to process a frame of data is negligible and that the acknowledgment frame is small compared to a data frame, the total time to send encoded data segmented into N_p frames is [19]

$$T_F = N_p\left(2t_{prop} + t_{frame}\right) \tag{4.4}$$

where t_{prop} is the propagation time from the transmitter to the receiver and t_{frame} is the length of a frame in seconds. If P_r is the probability that a single frame is in error, ACK and NACK frames are received error free and error detection is perfect, then the probability that it will take exactly k attempts to successfully transmit a frame is $P_r^{k-1}\left(1-P_r\right)$. The average number of retransmission is [21]

$$N_r = \sum_{k=1}^{\infty}\left(kP_r^{k-1}\left(1-P_r\right)\right) = \frac{1}{1-P_r} \tag{4.5}$$

The effective throughput for a channel with ARQ error control is the number of encoded data frames accepted by the receiver in the time it takes the transmitter to send a single data frame multiplied by the overhead factor of the transmitted packets [19]. Assuming that the bit transmission rate is R bps, the transmitter idle time can be expressed in terms of the number of bits ξ that could have been transmitted during the idle time [19]

$$\xi = R\left(t_{prop} + t_{proc} + t_{prop}\right) \quad \text{bits} \tag{4.6}$$

Each transmission involves sending an m-bit frame followed by an idle period ξ. The stop-and-wait scheme, on average, requires the transmission of $N_r\left(m+\xi\right)$ bits in order to send an n-bit block of data. Accordingly, the throughput for that system is

$$\eta_{ARQ} = \frac{n}{N_r\left(m+\xi\right)} = \frac{n}{m}\frac{\left(1-P_r\right)}{\left(1+\frac{\xi}{m}\right)} = r_{ARQ}\frac{\left(1-P_r\right)}{\left(1+\frac{\xi}{m}\right)} \tag{4.7}$$

where $r_{ARQ} = \frac{n}{m}$ is the rate of the error detecting code used as overhead in the design of the ARQ.

If the ACK frames are subjected to errors, the throughput expression changes to [19]

$$\eta = r_{ARQ}\left[\frac{\left(1-P_r\right)\left(1-P_{ACK}\right)}{\left(1+\xi/m\right)}\right] = \left(1-P_{ACK}\right)\eta_{error\ free\ ARQ} \tag{4.8}$$

where P_r is the probability of a bit error in the transmitted frame and P_{ACK} is the probability of error in the ACK frame.

From the aforementioned discussion, it is obvious that the retransmission of a frame increases the end-to-end delay. The total delay is given by [21]

$$D_F = T_F \frac{1}{1-P_r} \text{ sec} \tag{4.9}$$

C. HYBRID-ARQ PROTOCOLS

From the description of ARQ, as the channel quality deteriorates, there is an increase in the frequency of retransmission requests, which severely impacts the effective throughput. The hybrid ARQ protocols eliminate this effect through the use of forward error correction (FEC) in conjunction with error detection. The hybrid protocol can thus provide throughput similar to that of FEC systems while offering performance typical of ARQ protocols [19].

In hybrid-ARQ protocols, each information packet is encoded first with a forward-error-correction code and then with an error detection code. After the reception of the packet at the receiver, it is sent first to a FEC decoder and the resulting bitstream is sent to an error detection decoder. If errors are detected, a retransmission request is sent back to the transmitter; otherwise, the packet is accepted as correctly received.

The expression for effective throughput in the case of hybrid-ARQ transmission scheme is the same as Equation (4.7) of the ARQ only case with the exception that the probability of error and the coding rate are different [19]. Thus, the equation of effective throughput of hybrid-ARQ, given that the ACK frames are not subjected to errors, is [19]:

$$\eta_{HARQ} = r_{FEC} r_{ARQ} \frac{\left(1 - P_{DE} P_r\right)}{\left(1 + \frac{\xi}{m}\right)} \tag{4.10}$$

where P_{DE} is the probability of FEC decoder error and r_{FEC} is the code rate of the FEC scheme. The product of P_{DE} and P_r is the residual bit-error rate for the case of hybrid-ARQ. Average number of retransmissions for the hybrid-ARQ scheme is given by [19]

$$N_r = \frac{1}{1 - P_{DE} P_r} \tag{4.11}$$

Figure 4-3 shows the lower and upper bounds of the residual bit-error-rate for hybrid-ARQ in comparison with the bit-error-rate of pure ARQ for a binary symmetric channel. The FEC used for the hybrid-ARQ is a rate ½ convolutional encoder with a constraint length of 7. From Figure 4-3 and from Equations (4.10) and (4.11), it can be concluded that the hybrid-ARQ transmission scheme offers substantially better throughput performance than the pure ARQ. The drawback of the hybrid-ARQ is that for low bit-error rates, the effective throughput in the channel is lower because of the factor r_{FEC}.

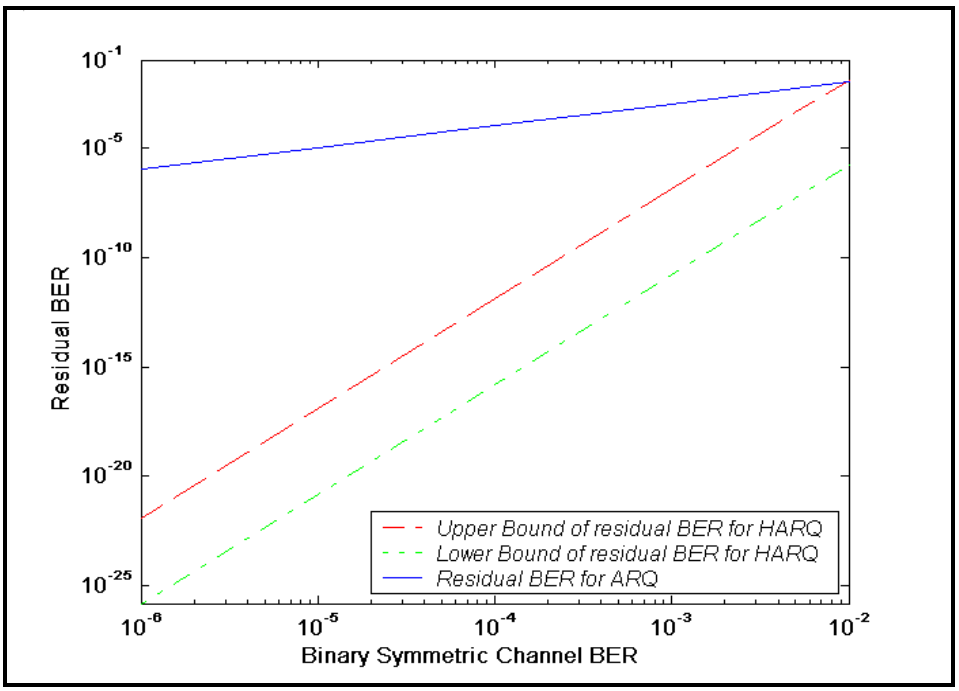


Figure 4-3. Upper and Lower Bounds of the residual BER for HARQ in Comparison with Residual BER of ARQ.

D. CHANNEL MODEL

A simple but effective approach to modeling a communication channel is by using a finite-state Markov model, where each state corresponds to a specific channel condition [22]. An error occurrence is not independent between bits or symbols as in memoryless channels, but is dependent on the errors introduced in the previously transmitted bits or symbols (channel with memory) [23]. The simplest such model is the Gilbert model, where the number of states is two: the good state corresponds to the total absence of errors and the bad state to error occurrence with a pre-defined probability profile. The advantage of using a Markov process to describe a channel is the ability to capture the changes in the quality of the medium that occur during the transmission and to simulate in this way the burst-error behavior of the channel. Also, it is possible to examine the performance of various transmission schemes for channels with memory.

1. Markov Models

Let Q denote a finite set of variables with dimension L. Let $\{X_0, X_1, ..., X_n, ...\}$ be a sequence of random variables, whose values are in Q. A discrete process is said to be a *homogeneous finite-state* k^{th}*-order Markov chain* if it satisfies the following conditions [22]

$$\begin{aligned} P\left(X_n = x_n \mid \{X_i = x_i\}_{i=0}^{n-1}\right) &= P\left(X_n = x_n \mid \{X_{n-m} = x_{n-m}\}_{m=1}^{k}\right) \\ &= P\left(X_k = x_k \mid \{X_m = x_m\}_{m=0}^{k-1}\right) \\ &\forall n > k,\ \{\mathrm{x_i} \in Q\}_{i=0}^{n-1} \end{aligned} \tag{4.12}$$

For Markov chains with order $m > 1$, the state transition probabilities are [22]

$$t_{k_1 \ldots k_{m+1}} = P\left(X_{m+1} = k_{m+1} \mid \{X_w = k_w\}_{w=1}^{m}\right) \tag{4.13}$$

In the first order Markov process or chain, transition probabilities are defined as follows [24]

$$t_{ij} = P\left(X_1 = j \mid X_0 = i\right) \tag{4.14}$$

where $i, j \in \{0,1\}$. For this case, the transition matrix is defined as [24]

$$T = \begin{bmatrix} t_{00} & t_{01} \\ t_{10} & t_{11} \end{bmatrix} \tag{4.15}$$

The idea behind using a number of states with a different error probability for each state is to represent different fading levels (quantized model) of the communication channel during transmission [23].

a. Fading Model

The effect of narrow-band fading on a baseband signal is [22]

$$\tilde{s}(t) = \tilde{u}(t)\tilde{f}(t) \ , \tag{4.16}$$

where $\tilde{u}(t), \tilde{f}(t), \tilde{s}(t)$ are the complex envelopes of channel input, fading and channel output, respectively. In order to obtain the finite state model, it is necessary to sample the analog fading process $\tilde{f}(t)$. The sampling period *T*, for this chapter's simulations, is chosen to be a symbol (8-bit) interval. Then, the instantaneous fading power $\alpha(t) = f^2(t)/2$ is quantized to a set of $L-1$ thresholds $\{A_k\}$ depending on the dimension of the state space that is chosen. The first and the last values of the above range of thresholds is $A_0 \triangleq 0$ and $A_L \triangleq +\infty$, respectively [22]. Figure 4-4 shows a simulated Rayleigh distributed signal envelope as a function of time (solid blue line). In order to quantize this signal into two states, we can set a threshold (such as the red dashed line in Figure 4-4) based on the desired channel error profile [23].

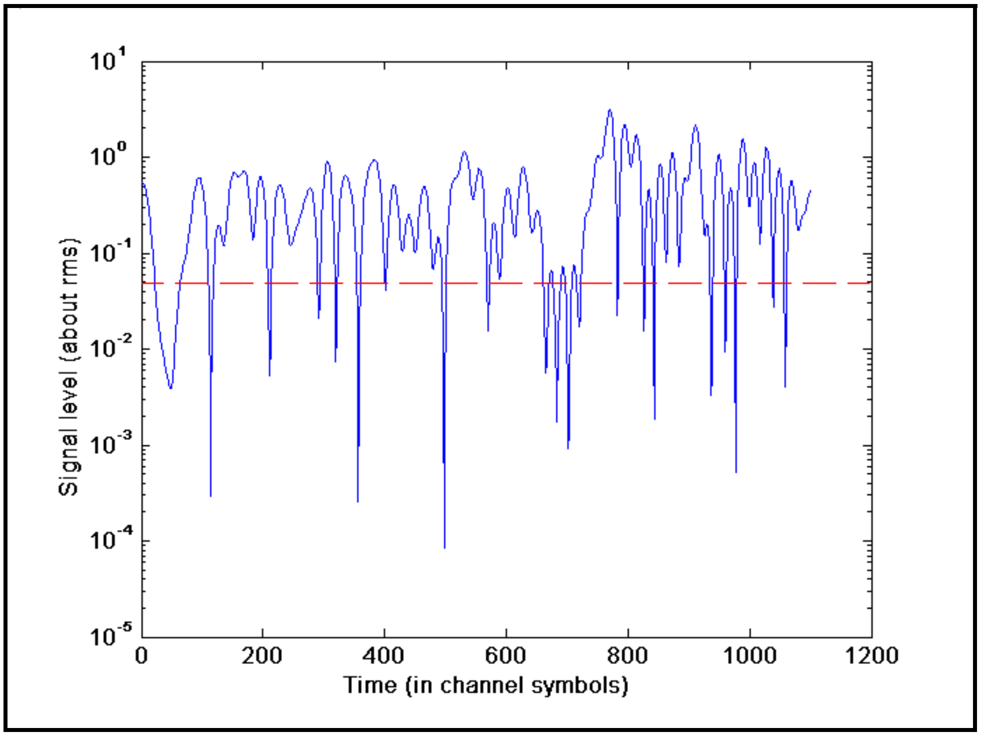


Figure 4-4. Quantization of Simulated Rayleigh Fading Signal in Two-State Markovian Process. Figure is taken from [23]. The Simulated Rayleigh Fading is Reproduced in MATLAB by using Jakes Model.

The general expressions of transition probabilities after the quantization of fading, assuming a slow fading channel (for example, $f_D T \ll 1$ where $f_D = \frac{v}{\lambda}$ is the maximum Doppler shift), follow the Markovian process and are given by [22]

$$\begin{aligned} t_{k,k+1} &\approx \frac{N_{K+1}T}{P_k}, & k &= 0,\dots,L-2 \\ t_{k,k-1} &\approx \frac{N_K T}{P_k}, & k &= 1,\dots,L-1 \\ t_{j,k} &\approx 0, & |\, j-k \,| &\geq 2 \\ t_{k,k} &= 1 - \sum_{j=0, j\neq k}^{L-1} t_{k,j}, & k &= 0,\dots,L-1 \end{aligned} \tag{4.17}$$

where N_k is the average crossing rate of the instantaneous fading power through level A_k and P_k is the probability of being in the state *k*.

First-order Markovian modes are more suited to approximate slow fading [22], [25], [23]. However, from a statistical analysis in [23], it was concluded that first-order models could represent an oversimplification if fading is not very slow (fading rate $f_D T \geq 10^{-2}$). The simplest quantized model is the two-state Markov model, which can be used in order to approximate the *block failure/success process* very well, and has already been proven in [22] and has been used for simulations in [24], [25], [26], [23] and [27].

b. Two-State Quantized Model

Figure 4-5 shows a state diagram for a quantized two-state Markov model called the Gilbert-Eliot model. It is a first order, discrete-time, stationary Markov chain model. The two states, good and bad, are denoted here for simplicity by G and B, respectively. The transition probabilities from the good state to the bad state and from the bad state to the good state are denoted by *b* and *g*, respectively.

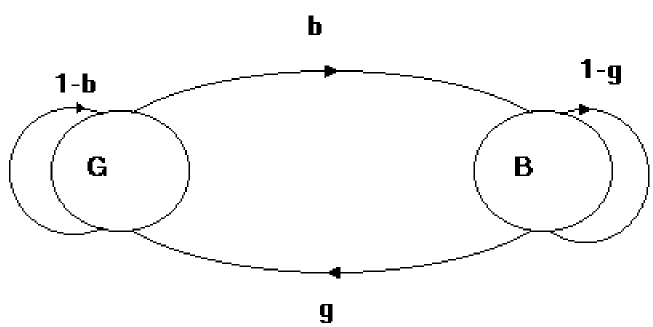


Figure 4-5. Two-State Markov Model.

The probability that the channel is in the good state or in the bad state at time τ is denoted by $P^{\tau}(G)$ and $P^{\tau}(B)$, respectively; or in matrix form: $P^{\tau} = \left[P^{\tau}(G), P^{\tau}(B)\right]$. Let $P^{\tau}(G \mid B)$ be the probability for the channel at time τ to be in state G given that at time 0 it was in state B. From Figure 4-5 and from Equation (4.15), the following transition matrix applies [23]:

$$T = \begin{bmatrix} 1-b & b \\ g & 1-g \end{bmatrix} \tag{4.18}$$

and

$$P^{\tau+1} = P^{\tau} T \tag{4.19}$$

Equations (4.18) and (4.19) indicate that how fast the channel changes from one state to the other is dependent on the values b and g. Large values of b and g imply a fast changing channel. For slow fading, as mentioned in [23], $b+g \ll 1$ is necessary. The value $(1-b)$ provides information concerning burst-error channel behavior. A higher value of $(1-b)$ signifies an increase in the probability of a transmitted packet remaining in the bad channel after one visit, thus leading to a longer burst-error for that packet. We set *(1-b)* = *ε* for simplicity. If P^{∞} is denoted as the stationary distribution for states G and B, then from [26], the probability for the channel to remain in the same state after a time interval T is given by the following equation [26], [23]

$$P^{\infty} = \left[P^{\infty}(G), P^{\infty}(B)\right] = \left[\frac{g}{b+g}, \frac{b}{b+g}\right] \tag{4.20}$$

Define the error probability as $P_e(G)$ and $P_e(B)$. If the time interval of channel changing is equal to a symbol duration, then the average symbol error rate

$$\begin{aligned} P_{avg} &= P_e(G)P^{\infty}(G) + P_e(B)P^{\infty}(B) \\ &= \frac{P_e(G)g + P_e(B)b}{b+g} \end{aligned} \tag{4.21}$$

In order to simulate a channel with the previously described behavior, it is necessary to know the probability that the channel is in a good state or bad state at time $\tau = m$ when the state at time $\tau = 0$ is given. These probabilities at $\tau = 0$ for good and bad states, respectively, are [23]

$$P^{m} = [1,0]T^{m} = \left[P^{\infty}(G) + P^{\infty}(B)(1-b-g)^{m}, P^{\infty}(B)\left(1-(1-b-g)^{m}\right)\right] \tag{4.22}$$

$$P^m = [0,1]T^m = \left[P^\infty(G)\left(1-(1-b-g)^m\right), P^\infty(B) + P^\infty(G)(1-b-g)^m \right] \qquad (4.23)$$

This channel model simulates conditions in which a transition to the bad state would cause a total corruption of the symbol or packet (burst errors). The run length of a burst has a geometric distribution with mean burst length $\frac{1}{g}$ [28].

E. NUMERICAL RESULTS AND SIMULATIONS

The image used for simulations is shown in Figure 4-15. It is an 8-bpp gray scale image with dimensions 900×1200. It consists of details, such as text and objects in the background, that can be affected by compression and during the transmission through a wireless channel. This image is compressed 8:1 to a bit resolution of 1 bpp (see Figure 4-16) using the JPEG2000 source code. The compressed image is then used for Monte Carlo simulations of transmission through a two-state wireless channel. The results of simulations presented here are based upon averages of 10 runs. Channel error probabilities are in the range of 10^{-6} to 10^{-1}.

The simulations are divided in two parts. In the first part, for different transmission schemes, the effective throughput is measured for three different capacity channels: 9.6-kbps (GSM), 64-kbps, and 1.5-Mbps. All channels have the same error probability. The transmission schemes used for comparison are the stop-and-wait ARQ, the rate ½ convolutional encoder with a constraint length of 7, and the hybrid-ARQ, which is a combination of the two previous schemes.

The second part of simulations examine the effectiveness of the JPEG2000 error resilient mechanisms. The channel behavior now is such that the above methods lead to very low effective throughput. The performance is evaluated by measuring the pSNR of the received image with and without error resilient tools. For this part, the same convolutional encoder as above is used to encode the bitstream.

For both parts of the simulation, three different packet sizes are used: 400, 800, and 1500 bytes. Simulations are repeated for three different burst-error channel behaviors with the steady-state probability of the channel being in a bad state ε taking values of 0.1, 0.01, and 0.001. A higher value of this steady-state probability gives longer burst-errors [22], [25], [29]. The bit-error-rate of the bad channel is chosen to be in the range between 0.5 and 10^{-2}, and for the good channel it is chosen to be between 10^{-4} and 10^{-7} in order to follow the Gilbert-Elliot model more closely. Finally, the channel is always stable for at least one symbol duration.

1. Simulation of Different Transmission Schemes

a. Effect of FEC on Effective Throughput

The selected image is compressed using no error resilient mechanisms. The compressed bitstream is saved with the extension *.jp2,* is read into *MATLAB* and is packetized. Each packet is sent independently through the proper encoder depending on the transmission scheme being simulated.

The FEC scheme uses a convolutional encoder as described previously followed by a block interleaver. The receiver first de-interleaves the received sequence and then decodes it. It orders the packets such that the resulting sequence is readable by the JPEG2000 encoder and saves the result as an unsigned integer file with the extension *.jp2*. The JPEG2000 decoder then decompresses this file and extracts the received image. The rate ½ convolutional encoder gives an effective throughput of 0.5 for the entire range of the average BER independent of the packet size used. The received image is not error free and the performance is measured in terms of the pSNR of the received image. The red cross points of Figures 4-13 through 4-15 represent the pSNR of the received image. Degradation of pSNR occurs when the average BER of the channel increases or when the steady-state probability for a packet to remain in the bad (ε) increases.

b. Effect of Stop-and-Wait ARQ on Effective Throughput

No coding takes place in the stop-and-wait ARQ protocol. After packetizing the compressed sequence, a 60 bytes header is added to each packet by the *MATLAB* code. The receiver examines each packet for errors. Assuming perfect error detection and error free reception of ACKs, the receiver sends a negative ACK to the transmitter and requests retransmission in case an error is detected. After the error free reception of all the transmitted packets, the receiver removes the header of each packet, organizes them in the correct order and saves the resulting sequence as an unsigned (8 bit precision) file with extension *.jp2*. The JPEG2000 decoder is then used to extract the image from this sequence.

In order to evaluate the stop-and-wait ARQ performance, the effective throughput is calculated using Equations (4.5) through (4.7). The distance between the transmitter and the receiver is set to 10 km and is used in order to estimate the propagation time t_{prop} of Equation (4.6).

The range of average BER for the two-state quantized channel is chosen to be sufficient in order to estimate the values at which the resulting effective throughput becomes unacceptable. A symbol interval of 8 bits is chosen for the sampling period of Equation (4.22), which indicates that the channel is always stable at least for one symbol duration.

Figures 4-6 through 4-8 display the affect of packet size and ε on effective throughput for the stop-and-wait ARQ scheme. The steady-state probability of the channel being in bad state ε is displayed in the lower left corner of each figure. For 9.6-kbps and 64-kbps channels, the effect of the transmitted packet size on the effective throughput is not significant for average channel BERs below 10^{-5}. For greater BERs, the smaller packet size achieves better effective throughput. For the 1.5-Mbps channel, because of the higher channel capacity, the use of small packets causes a significantly low effective throughput even for low average BERs. For this channel capacity, the use of a large packet size is necessary. For all channels, when $\varepsilon = 0.1$, the effective throughput drops rapidly for BERs above 10^{-5}. Thus, the average number of transmissions and the resulting transmission delay increase rapidly. For BERs above 10^{-5}, it is best to use another transmission scheme. Results for lower values of ε are included in Appendix B.

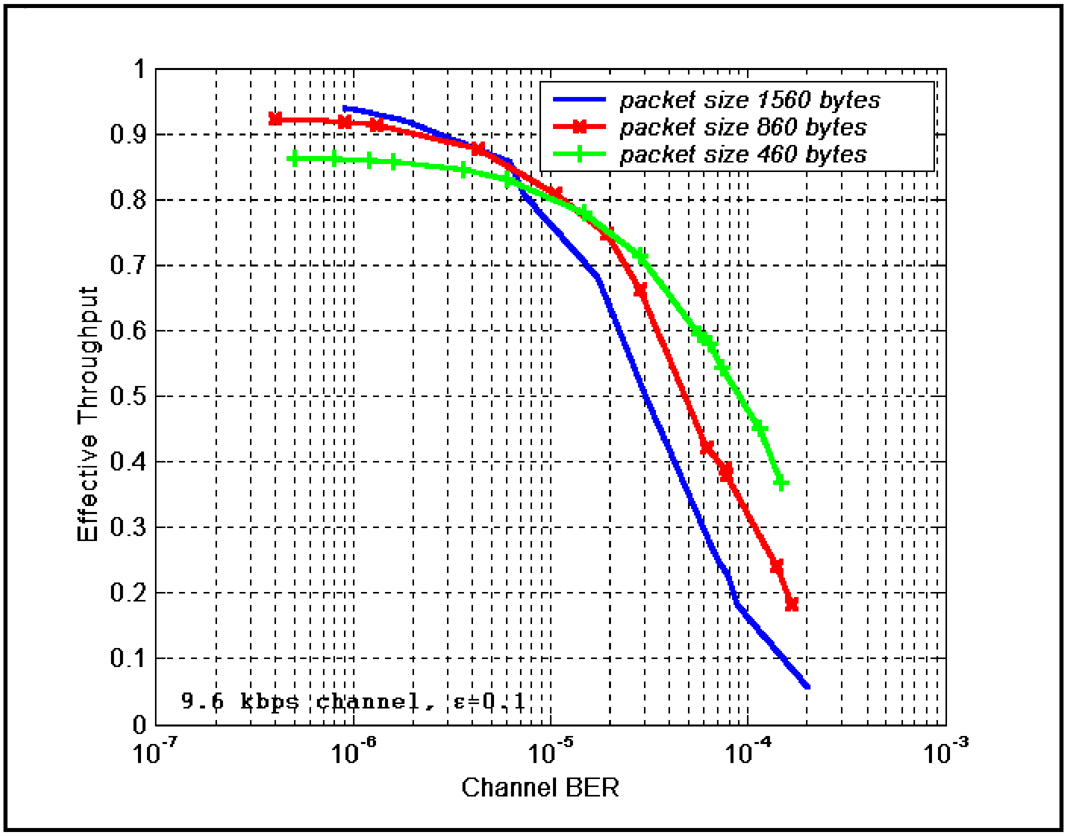


Figure 4-6. Stop-and-Wait ARQ: Effective Throughput for a 9.6-kbps (GSM) Channel with $\varepsilon = 0.1$.

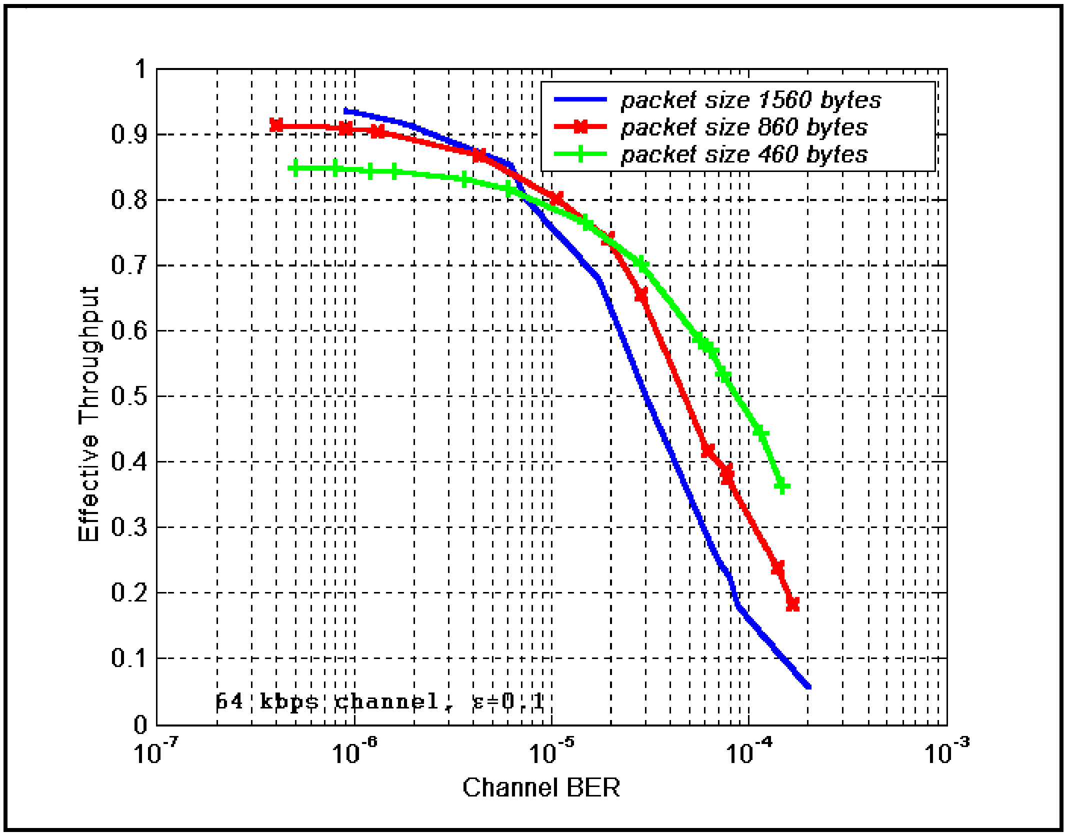


Figure 4-7. Stop-and-Wait ARQ: Effective Throughput for a 64-kbps Channel with $\varepsilon = 0.1$.

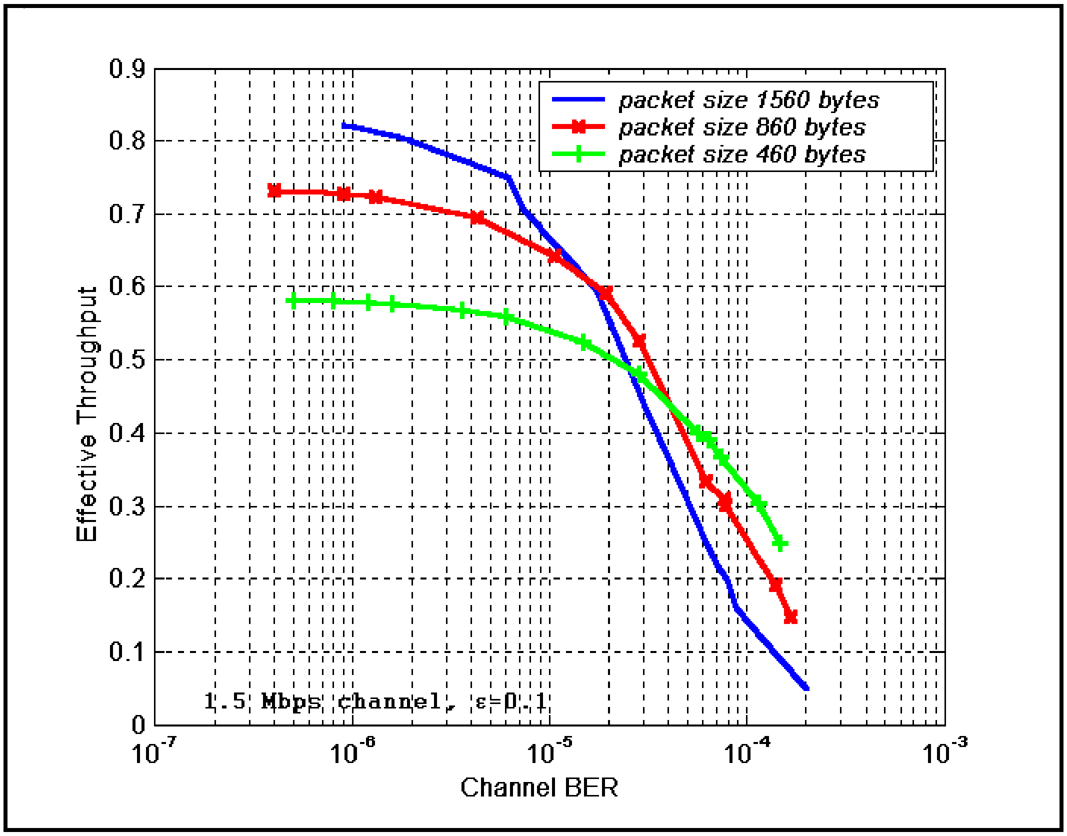


Figure 4-8. Stop-and-Wait ARQ: Effective Throughput for a 1.5-Mbps Channel with $\varepsilon = 0.1$.

c. Effect of Hybrid-ARQ on Effective Throughput

For hybrid-ARQ, a convolutional encoder and a block interleaver are used prior to compression. A 60-byte header is added to each coded packet. The receiver sends each packet through a deinterleaver immediately after reception and then through a Viterbi decoder. The output of the decoder is examined for uncorrected or undetected errors. If any occur, the receiver then sends a negative ACK to the transmitter and requests retransmission of that packet.

Figures 4-9 through 4-11 display the affect of packet size and the channel parameter ε on effective throughput for the hybrid-ARQ scheme. Similar to the pure ARQ scheme, there is no significant difference in effective throughput between a 9.6-kbps and a 64-kbps channel. The packet size is an important parameter of effective throughput for the 1.5-Mbps channel. The performance for $\varepsilon = 0.1$ is more stable when compared to the stop-and-wait ARQ scheme for channel conditions noisier than 5×10^{-5}. When the channel average BER becomes greater than 4×10^{-4}, the effective throughput drops dramatically. The results for lower values of ε are included in Appendix B.

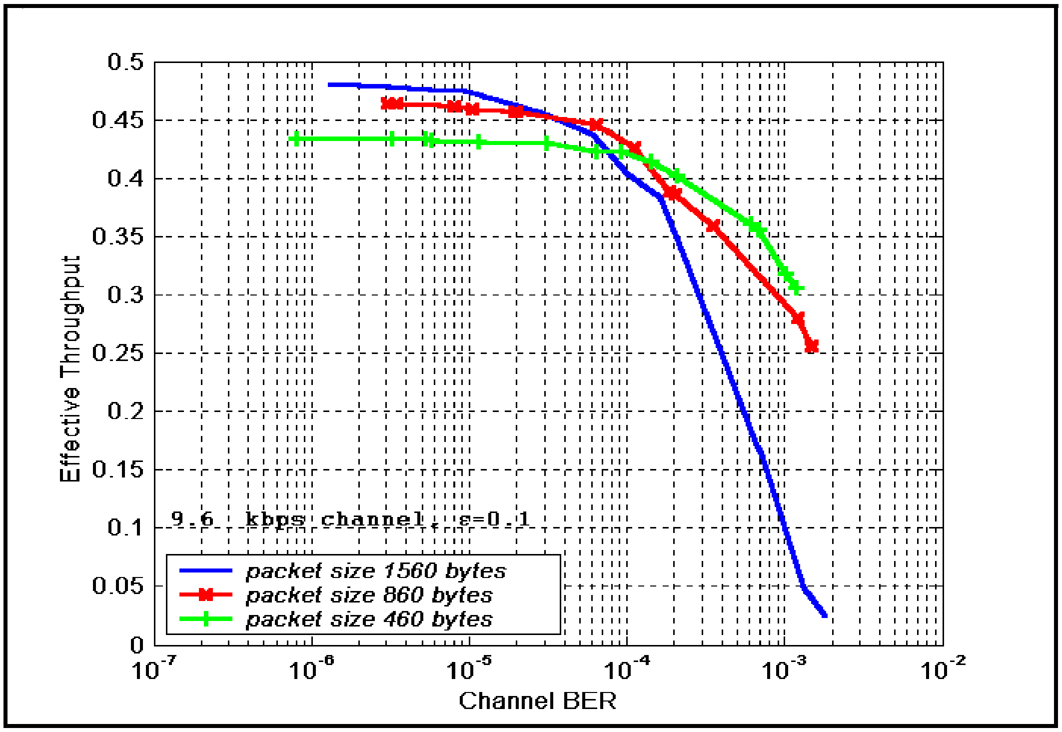


Figure 4-9. Hybrid-ARQ: Effective Throughput for a 9.6-kbps (GSM) Channel with $\varepsilon = 0.1$.

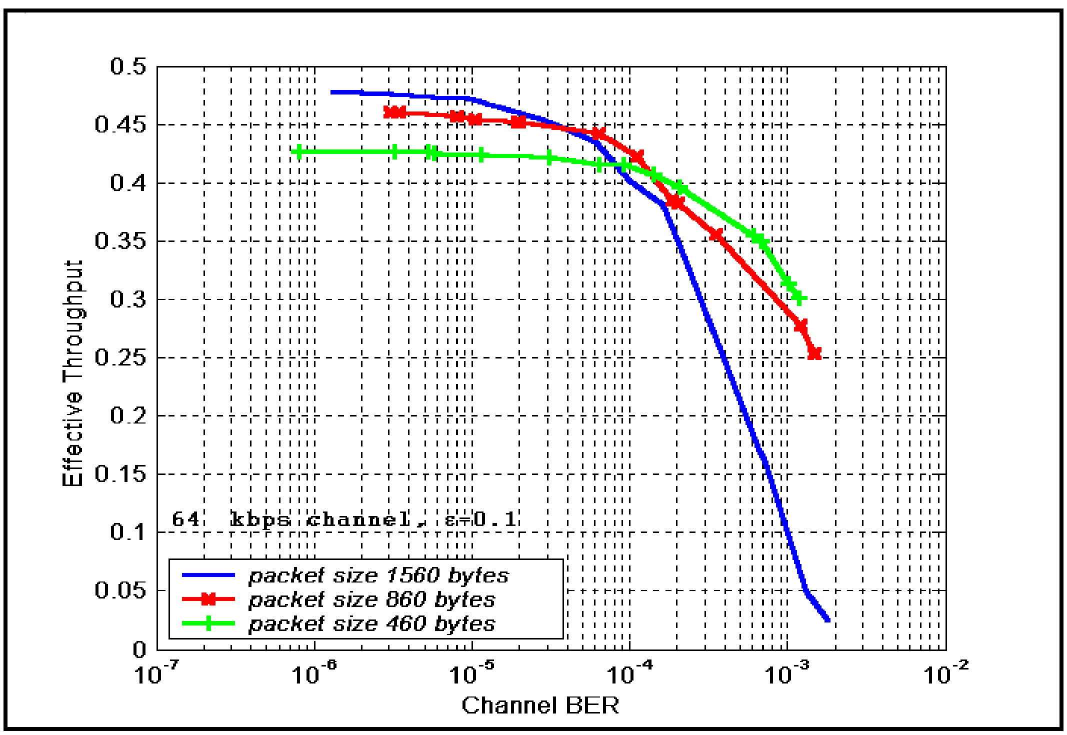


Figure 4-10. Hybrid-ARQ: Effective Throughput for a 64-kbps Channel with $\varepsilon = 0.1$.

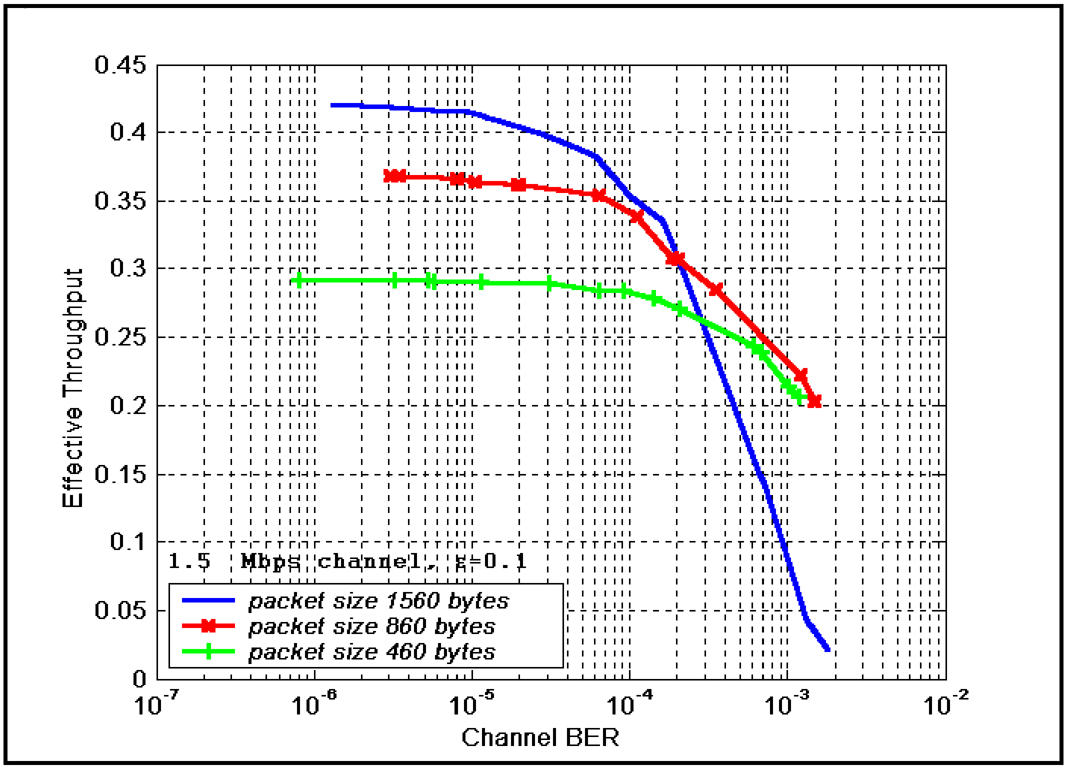


Figure 4-11. Hybrid-ARQ: Effective Throughput for a 1.5-Mbps Channel with $\varepsilon = 0.1$.

2. JPEG2000 Error Resilient Mode

This part of the simulations evaluates the performance of error resilient mechanisms of the JPEG2000 still image compression standard. The channel behavior for these simulations is not in the range (10^{-7} to 10^{-2}) of the previous simulations but is restricted to a range (10^{-4} to 10^{-2}) where considerable reduction of effective throughput was obtained while using the above schemes. The error resilient mechanisms used are those mentioned in Chapter III. The rate ½ convolutional encoder with constraint length of 7 adds redundancy to the transmitted sequence of packets. The simulations are also repeated for JPEG2000 compressed images without error resilient tools in order to compare the respective results. After the compression and before transmission, the pSNR of the image without error resilient tools is 37.078 dB and with error resilient tools, it is 36.906 dB. For each case, the simulation takes into account the size of the packet and the steady-state probability of a packet to be in the bad channel (ε).

Figures 4-12 through 4-14 show the performance of JPEG2000 on a compressed image segmented at various packet sizes with and without error resilient mechanisms and transmitted through the simulated channel with $\varepsilon = 0.1$. From these figures, it can be concluded that the error resilient features of JPEG2000 significantly enhance the image quality. This observation is reinforced under high channel noise conditions as well. The improvement when error resilient mechanisms are used appears to be in the range of 3 to 10 dB.

Additionally, a result that cannot be displayed in the figures is that, without error resilient mechanisms, about 15% of the received images could not be decoded due to one of the following reasons:

- Input codestream does not commence with a start of sequence (SOC) marker
- The upper left hand of the image has been displaced so far from the origin of the canvas coordinate system that the first tile of the image is completely empty
- Input codestream does not appear to contain an initial start of tile (SOT) marker

This number drops to 1% when error resilient tools are used.

Figures 4-15 and 4-16 show the original and the compressed image. Figures 4-17 and 4-18 show the received image without and with error resilient tools, respectively, for $\varepsilon = 0.1$ and average BER10^{-3}. Appendix B shows additional results for $\varepsilon = 0.01$ and $\varepsilon = 0.001$.

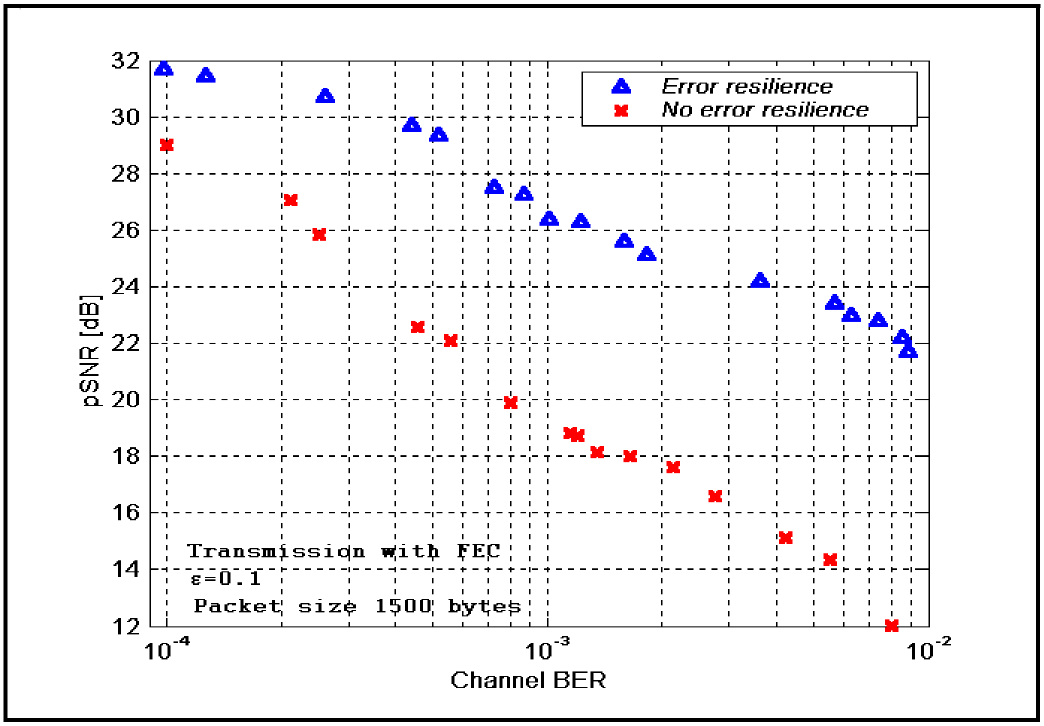


Figure 4-12. Performance of JPEG2000 With and Without Error Resilient Tools. The Packet size Used for Transmission is 1500 Bytes.

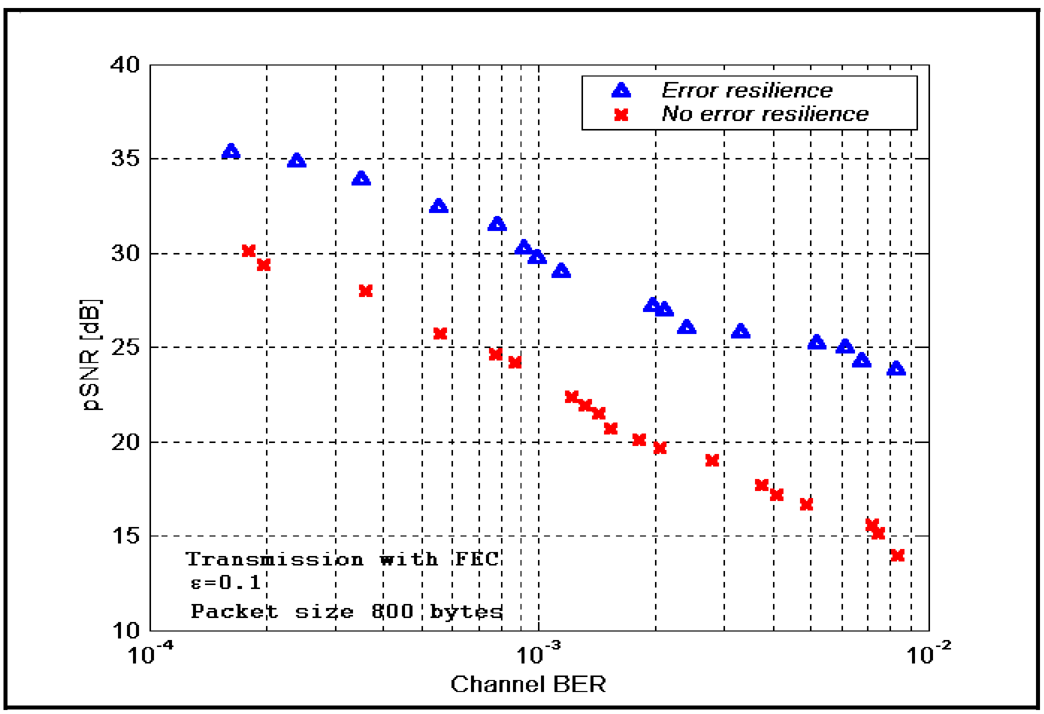


Figure 4-13. Performance of JPEG2000 With and Without Error Resilient Tools. The Packet size Used for Transmission is 800 Bytes.

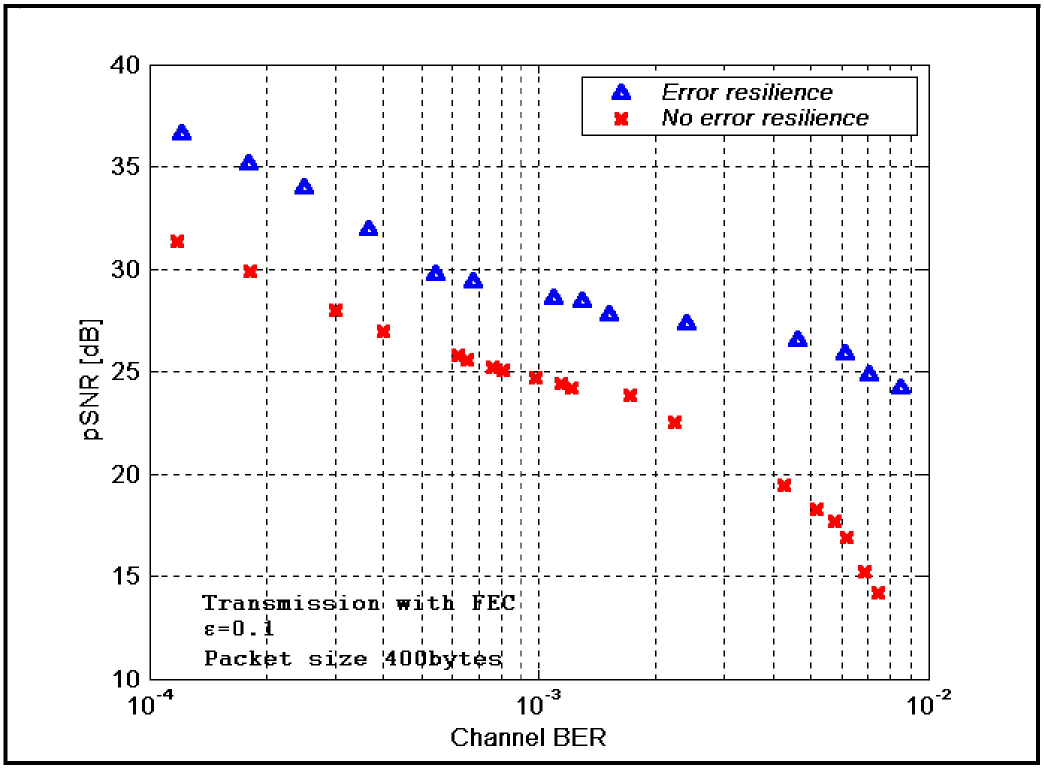


Figure 4-14. Performance of JPEG2000 With and Without Error Resilient Tools. The Packet size Used for Transmission is 400 Bytes.

Figure 4-15. Original Image.

Figure 4-16.   Compressed Image (8:1).

Figure 4-17. Received Image without Error Resilient Tools (pSNR 19.20 dB).

Figure 4-18. Received Image with Error Resilient Tools (pSNR 26.40 dB).

F. SUMMARY

The first part of this chapter described three basic data transmission schemes: forward error correction, stop-and-wait ARQ and hybrid-ARQ. Wireless channels are modeled using the Gilbert-Eliot model, which is based on a two-state Markov process. In all cases, the discussion is supported by simple analysis.

Experiments performed in this chapter include investigating the affects of these transmission schemes, the size of the compressed image packets and the channel behavior on the effective throughput of the channel. The error resilient tools of JPEG2000 code were used to enhance the image quality for specified channel conditions

V. CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE WORK

The objective of this thesis was to investigate the performance of the JPEG2000 still image compression standard and examine its error resilient mechanisms under various constraints of a bandlimited, noisy channel. JPEG2000 source code and JPEG source code were used to compare their performance. A wireless channel based on a two-state Markov chain is modeled and simulated in conjunction with compressed image transmission.

A. CONCLUSIONS

The JPEG still image compression scheme has provided poor compression for grayscale images at bit resolutions lower than 0.5 bpp. At low bit resolutions, distortion is high and the subjective image quality is poor due to blockiness and ringing artifacts. In comparison, the JPEG2000 still image compression standard has provided higher compression rates (better than 80:1), with lower distortion and better image quality. JPEG2000 provides features, such as region of interest coding and lossless compression, that are not available in the baseline JPEG. The superior performance of JPEG2000 over JPEG however is at the expense of the complexity of the algorithm. Consequently, JPEG2000 is recommended for applications that require high compression rates while JPEG is appropriate for low complexity applications.

Both compression schemes have been investigated for image transmission over bandwidth-limited, noisy channels. The bitstream of each compression scheme was encoded using three different error control schemes: FEC, ARQ and hybrid-ARQ. The baseline JPEG bitstream was found to be unreliable for image transmission over noisy channels due to frequent loss of synchronization between the bitstream and the decoder. In comparison, JPEG2000 provides various error resilient mechanisms that enable the decoder not only to achieve synchronization with the bitstream, but also to detect and correct errors that were injected into the bitstream during transmission. The received image quality with error resilient tools is superior to that of JPEG or JPEG2000 without error resilient tools; the improvement is in the range of 3 to 10 dB.

Image compression with a specified region of interest using JPEG2000 has also been examined. This feature of JPEG2000 enables the user to define regions of interest of any shape and size and code the selected regions at a better quality than the rest of the image. The effectiveness of the region of interest feature is demonstrated using several images and for different shapes.

B. RECOMMENDATIONS FOR FUTURE WORK

The Gilbert-Eliot channel based on a two-state Markov chain was used in this thesis. An extension of this work may consider more accurate channel models. For example, a model based on a four-state Markov chain, which can simulate two-state communication channels with two-state service rates (queuing system) for ARQ protocols such as go-back-N is of interest [25].

For image transmission over simulated wireless channels, the forward error correction technique used in this thesis was the convolutional code. A future effort may investigate the use of other forward error correction schemes, such as turbo codes or Reed-Solomon codes, which may improve the image quality.

In this work, the compressed image transmission was limited to point-to-point wireless channels. Investigation of image data transmission over multi-node networks is recommended along with an evaluation of error resilient mechanisms of JPEG2000 under network congestion conditions.

APPENDIX A.

A. PROGRESSIVE BY RESOLUTION AND BY SNR TRANSMISSION

The structure of the bitstream based on the packets and their organization in layers is responsible for the image reception by the decoder. The received image may be a single layer bitstream organization, a multi-layer “*resolution progressive*” bitstream organization, or a multi-layer “*SNR progressive*” bitstream organization. Figure A-1 (a)-(d) and Figure A-2 (a)-(c) illustrates this process. As bitstreams corresponding to higher layers are received and added to bitstreams from previous layers, the quality of the image improves and the size increases.

(a). Level one of progressive by resolution transmission

(b). Level two of progressive by resolution transmission

(c). Level three of progressive by resolution transmission

(d). Level four of progressive by resolution transmission

Figure A-1. Levels of Progressive by Resolution Transmission.

(a). Level one of progressive by SNR transmission

(b). Level two of progressive by SNR transmission

(c). Level three of progressive by SNR transmission

Figure A-2. Levels of Progressive by SNR Transmission.

B. PERFORMANCE COMPARISON BETWEEN JPEG2000 AND JPEG

In order to demonstrate the superior performance of JPEG2000 over JPEG for image compression, the image "*Building*" was used. JPEG is not able to compress the image for bit resolutions less than 0.15 bpp and the compressed image quality is unacceptable (by visual evaluation) due to the blocking artifacts. JPEG2000 can compress the images with acceptable quality for bit resolutions less than 0.065 bpp. Figures A-3 through A-5 provide results of comparison.

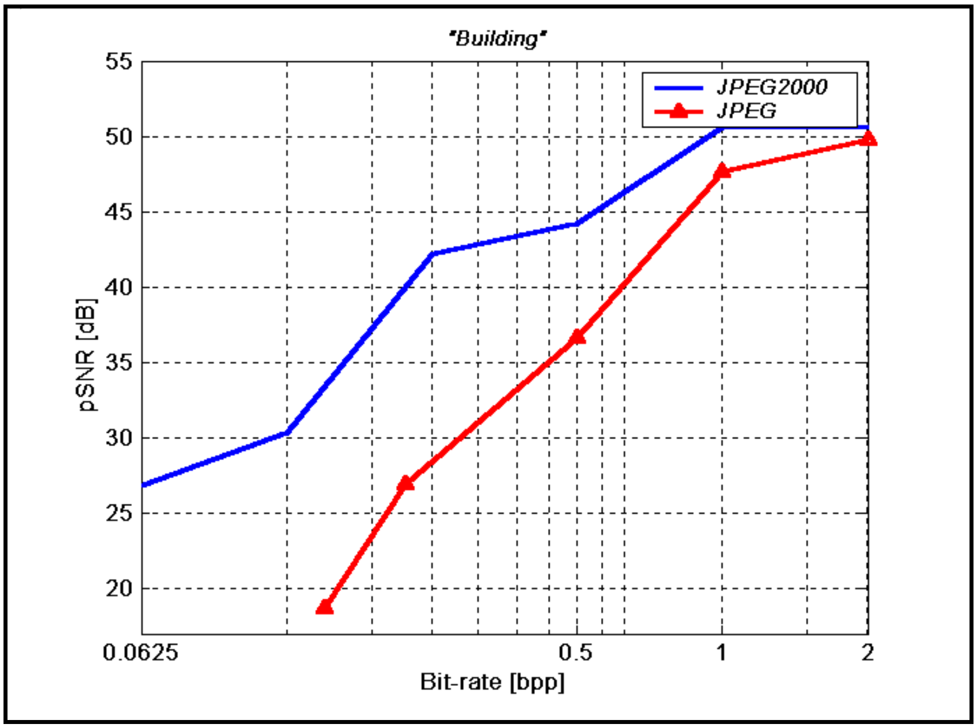


Figure A-3. Compression Performance of JPEG2000 for the Image "*Building*" in Comparison with the JPEG.

Figure A-4. JPEG2000 Compressed Image with Bit resolution 0.0625 bpp.

Figure A-5. JPEG Compressed Image with Bit resolution 0.15 bpp.

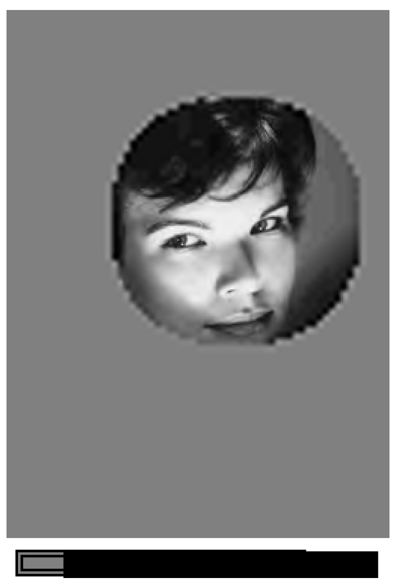
C. EXAMPLES OF REGION OF INTEREST CODING IN JPEG2000

Figure A-6 illustrates an example of circular ROI. The image "Building" compressed at an average bit resolution of 0.25 bpp in a way that the selected region of interest has a higher fidelity than the rest of the image.

Figure A-6. Example of Circular ROI and Bit resolution 0.25 bpp for the Image "*Building*".

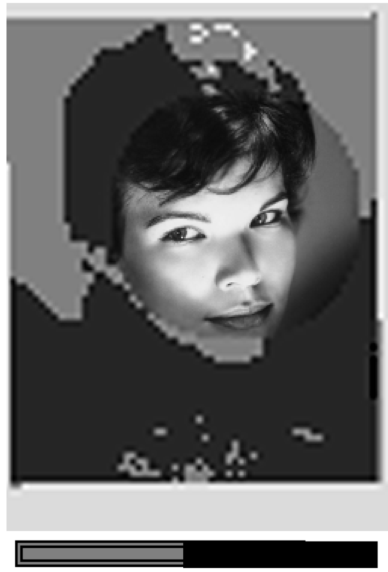
During the embedded coding process, the coefficient bits of the ROI are placed in the bitstream before the background parts of the image. Thus, the ROI is decoded before the rest of the image. Regardless of the scaling, a full decoding of the bitstream results in a reconstruction of the whole picture with the highest fidelity available. In order to simulate the progressive decoding of an image transmitted with a region of interest, the image "*Woman*" (see Figure A-7) is compressed at different bit resolutions with the same circular ROI. Figure A-8 (a)-(d) present these results.

Figure A-7. Original Image “*Woman*”.



0.0125 bpp

(a). Decoding Process: Level One of Image With Region Of Interest



0.25 bpp

(b). Decoding Process: Level Two of Image With Region of Interest

0.50 bpp

(c). Decoding Process: Level Three of Image With Region of Interest

2bpp
(d). Decoding Process: Level Four of Image With Region of Interest

Figure A-8. Levels of Decoding Process of Image With Region of Interest.

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX B.

Bursty channel behavior is simulated in order to determine the effectiveness of the error resilient tools of JPEG2000.

Figures B-1 and B-2 show the effective throughput results for 9.6-kbps (GSM) channel with $\varepsilon = 0.01$ and $\varepsilon = 0.001$, respectively, by using the ARQ transmission scheme. Effective throughput is reduced for any packet size for channel BERs above 7×10^{-5}.

Figures B-3 and B-4 show the effective throughput for the hybrid-ARQ scheme for the same values of ε as above. The effective throughput for hybrid ARQ is very low for channels with average bit-error-rates lower than 5×10^{-4}. Error resilient tools are required for channel BERs above 5×10^{-4}.

Figures B-5 through B-10 show the performance of JPEG2000 error resilient tools. The compressed bitstream is coded additionally with rate ½ convolutional encoder with a constraint length of 7. For average channel BERs above 10^{-5}, error control schemes like ARQ and hybrid-ARQ lead to low effective throughput. The steady state probability ε takes the values of 0.01 and 0.001. The improvement in performance with error resilient mechanisms is in the range of 2 to 10 dB compared to without error resilience.

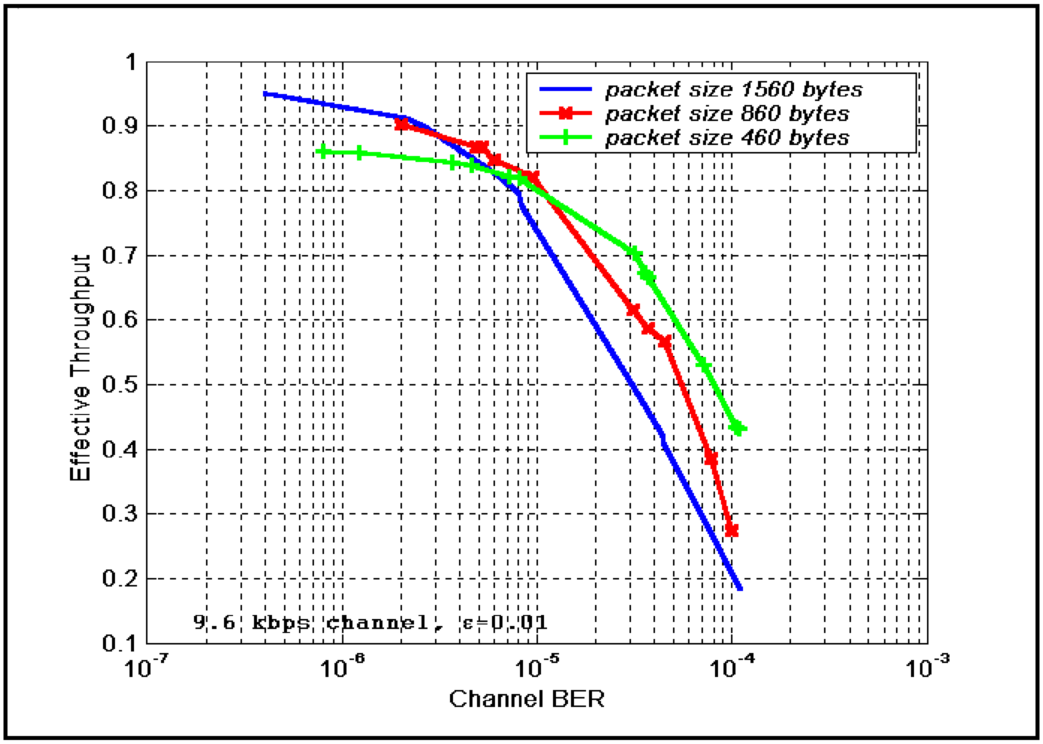


Figure B-1. Stop-and-Wait ARQ: Effective Throughput for a 9.6-kbps (GSM) Channel with $\varepsilon = 0.01$.

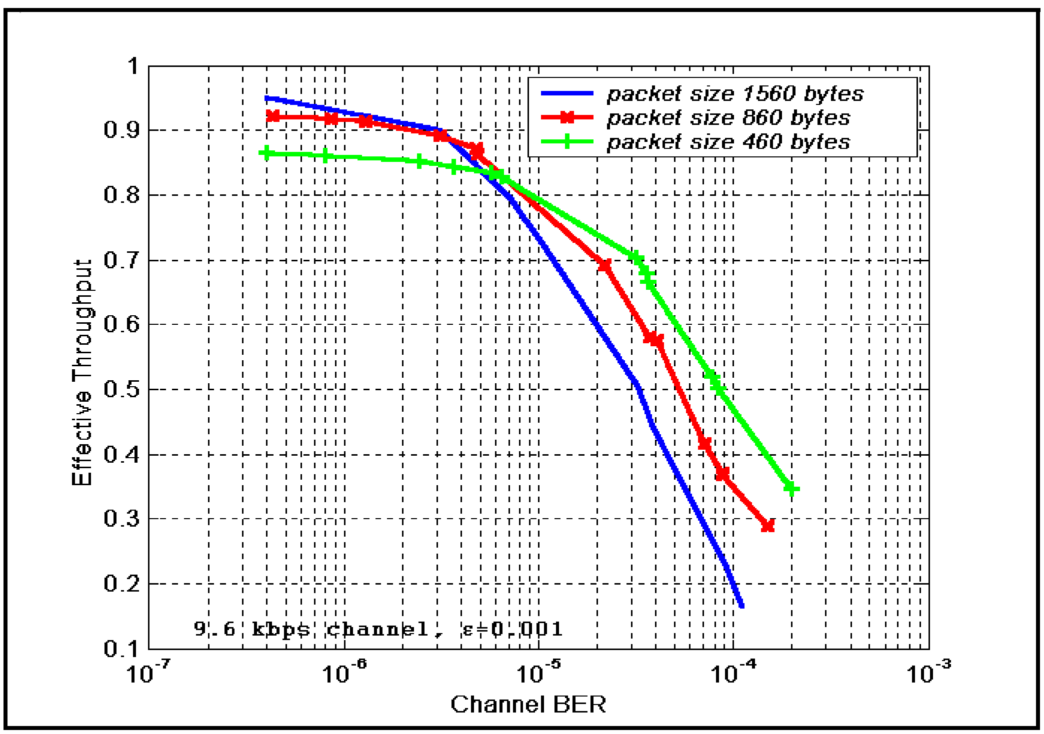


Figure B-2. Stop-and-Wait ARQ: Effective Throughput for a 9.6-kbps (GSM) Channel with $\varepsilon = 0.001$.

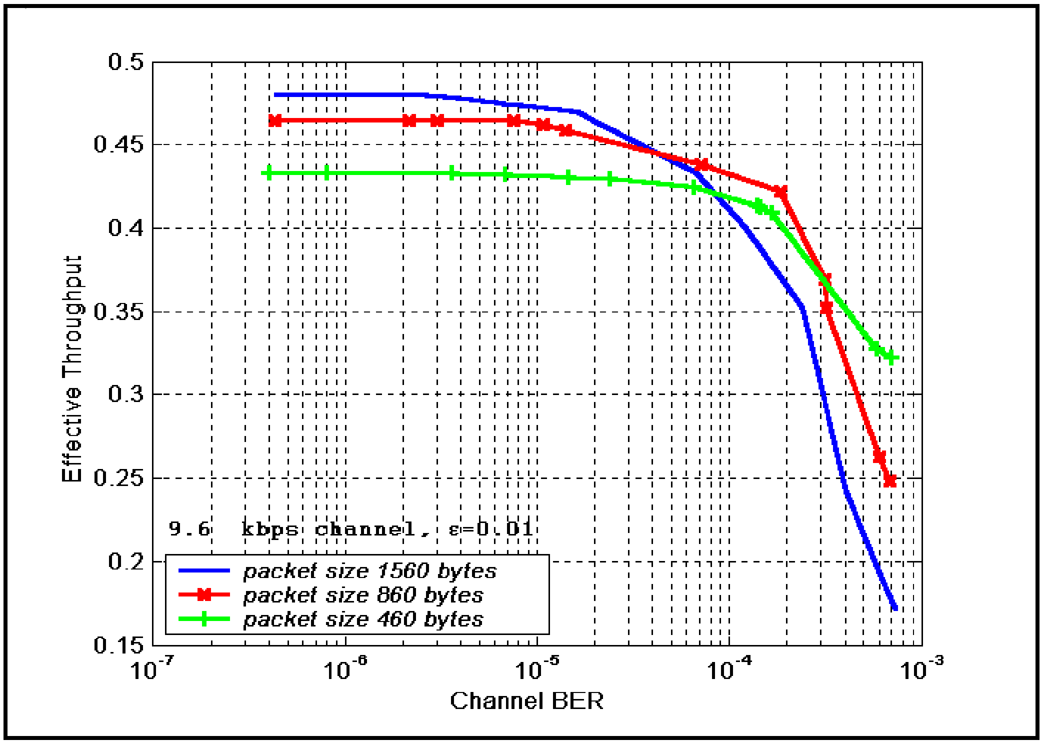


Figure B-3. Hybrid-ARQ: Effective Throughput for a 9.6-kbps (GSM) Channel with $\varepsilon = 0.01$.

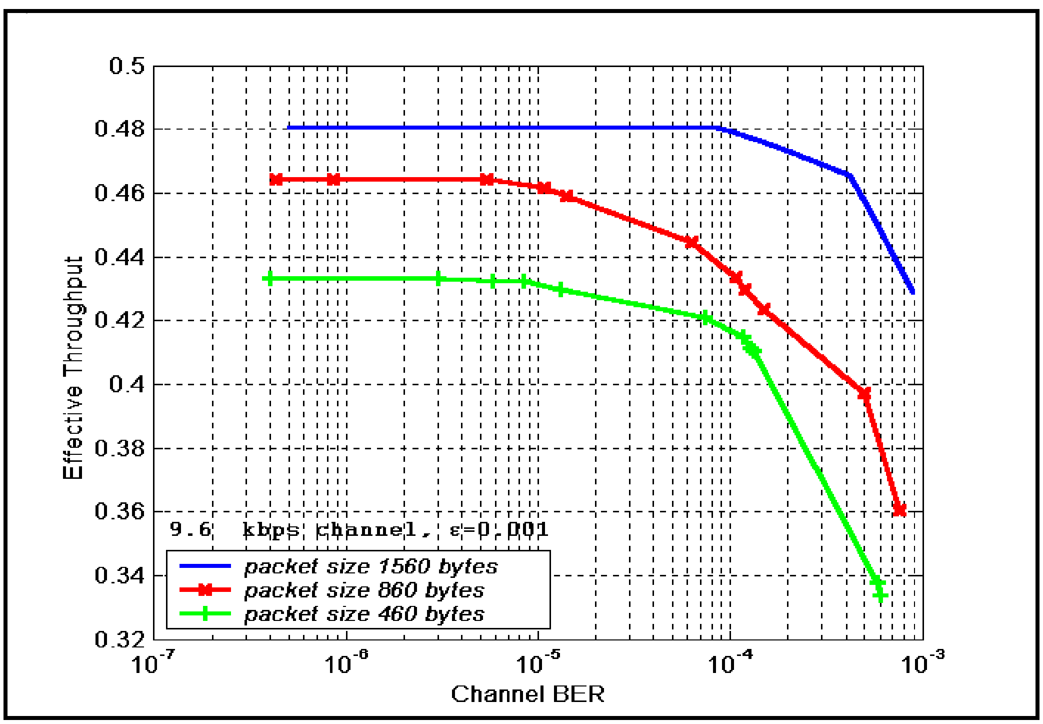


Figure B-4. Hybrid-ARQ: Effective Throughput for a 9.6-kbps (GSM) Channel with $\varepsilon = 0.001$.

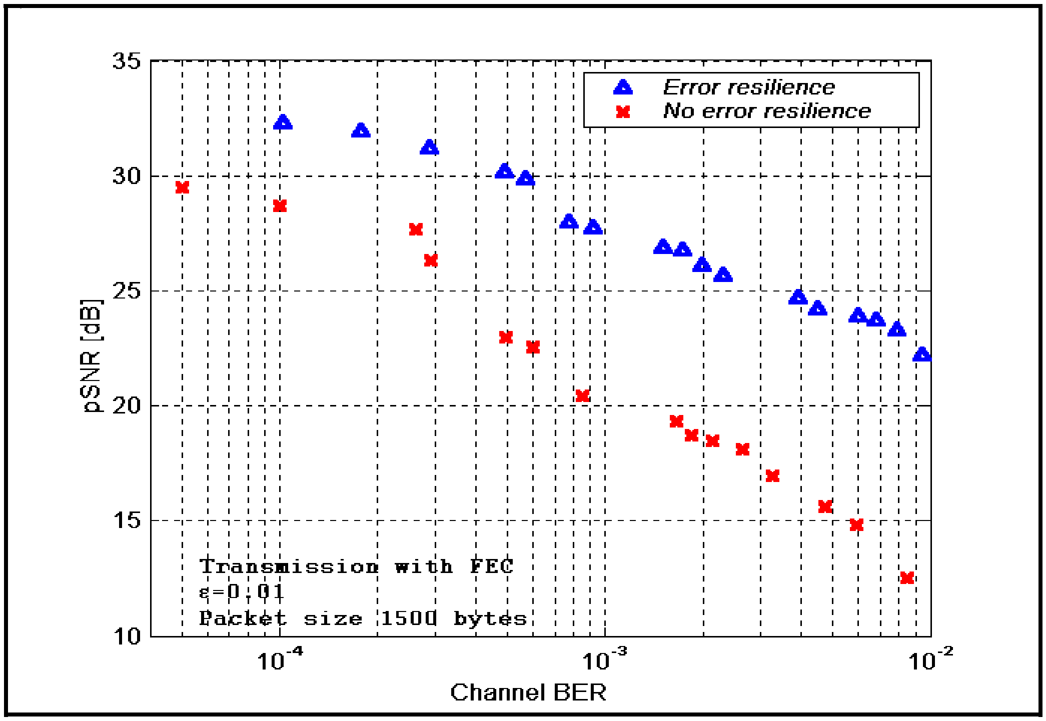


Figure B-5. Performance of JPEG2000 With and Without Error Resilient Tools. The Packet size Used for Transmission is 1500 Bytes.

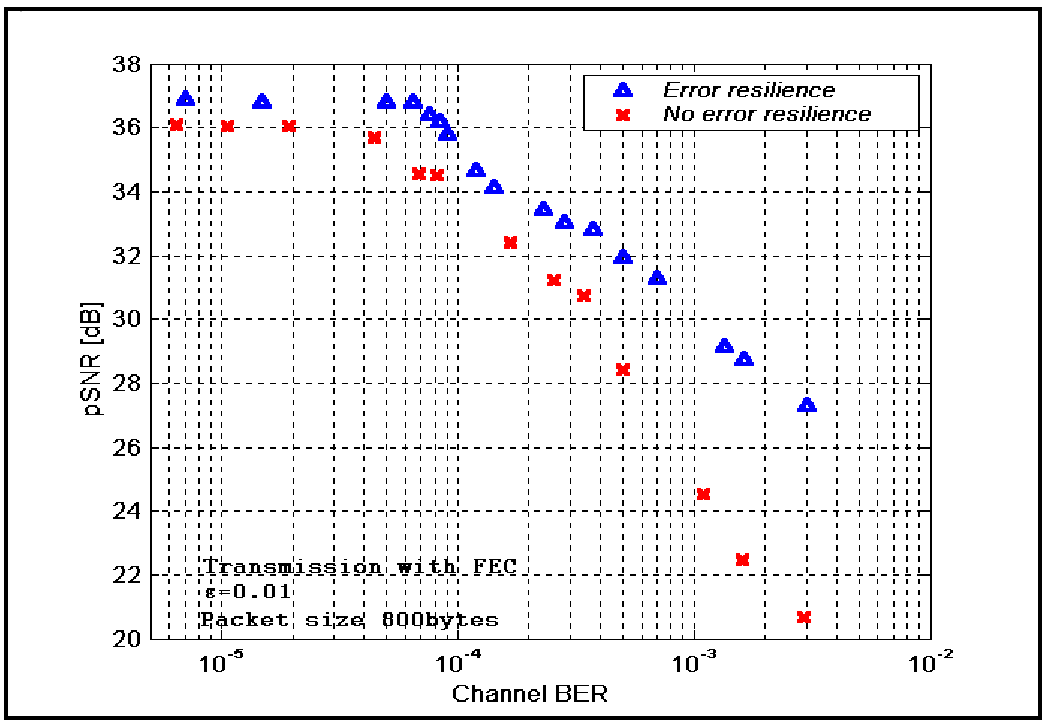


Figure B-6. Performance of JPEG2000 With and Without Error Resilient Tools. The Packet size Used for Transmission is 800 Bytes.

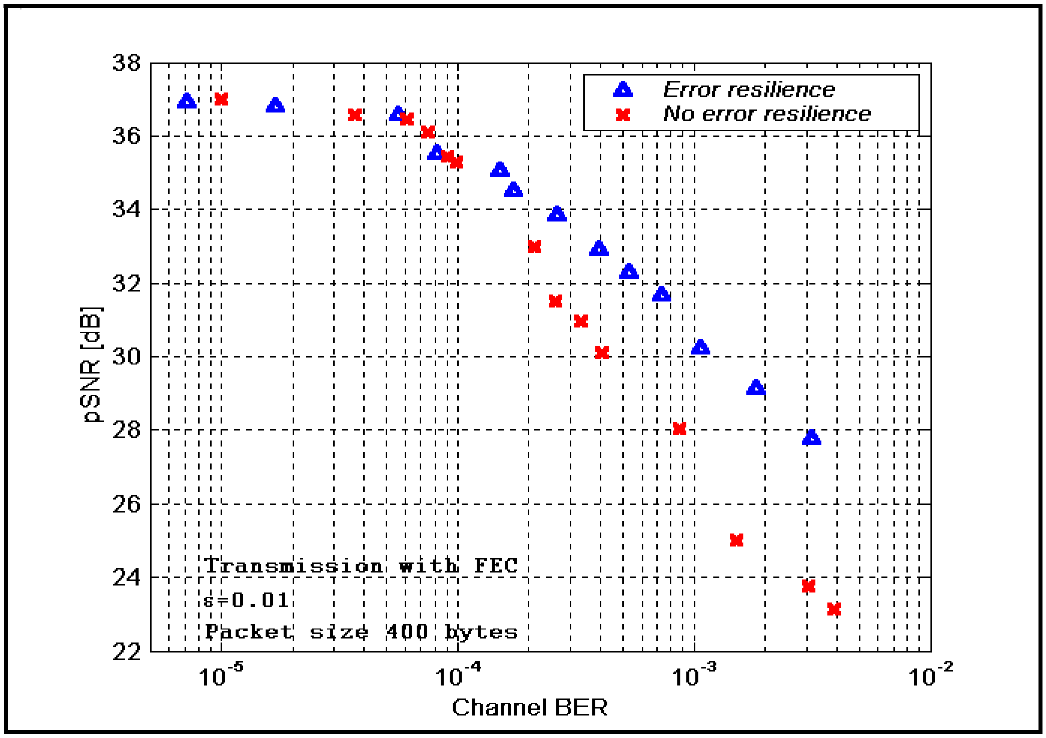


Figure B-7. Performance of JPEG2000 With and Without Error Resilient Tools. The Packet size Used for Transmission is 400 Bytes.

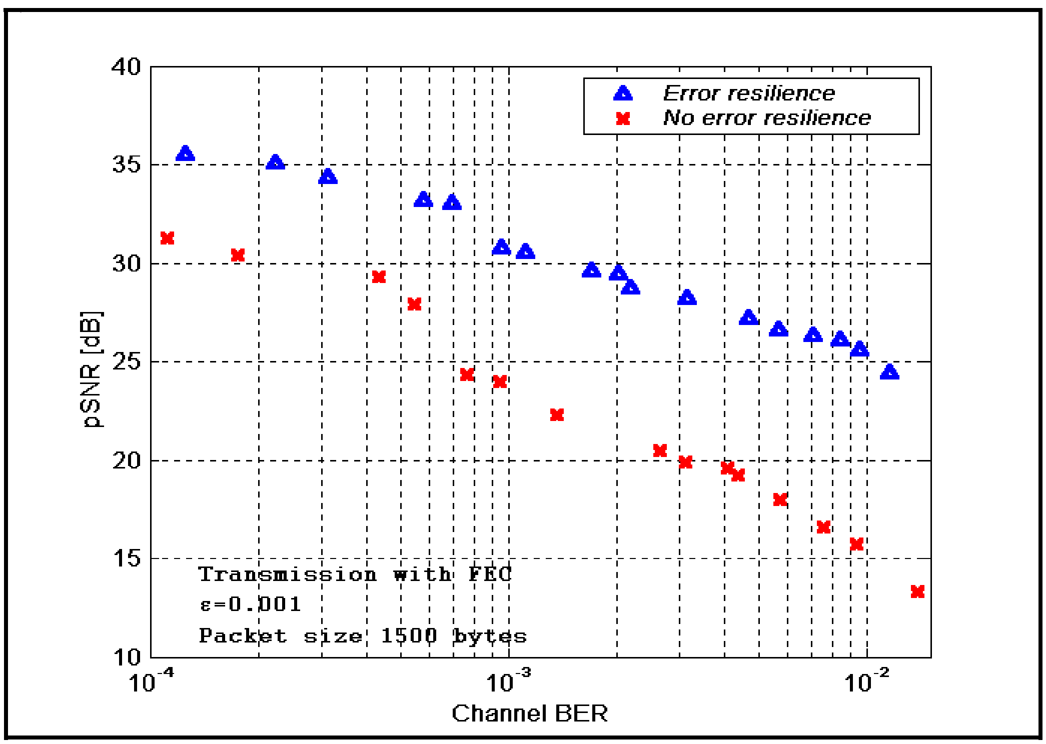


Figure B-8. Performance of JPEG2000 With and Without Error Resilient Tools. The Packet size Used for Transmission is 1500 Bytes.

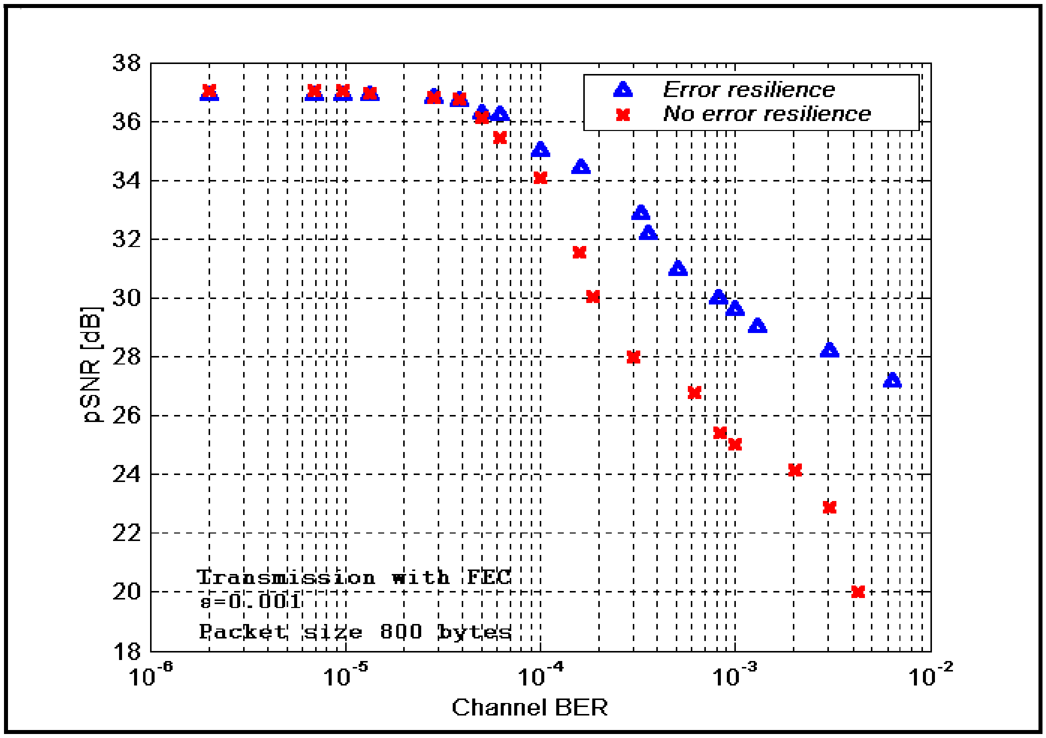


Figure B-9. Performance of JPEG2000 With and Without Error Resilient Tools. The Packet size Used for Transmission is 800 Bytes.

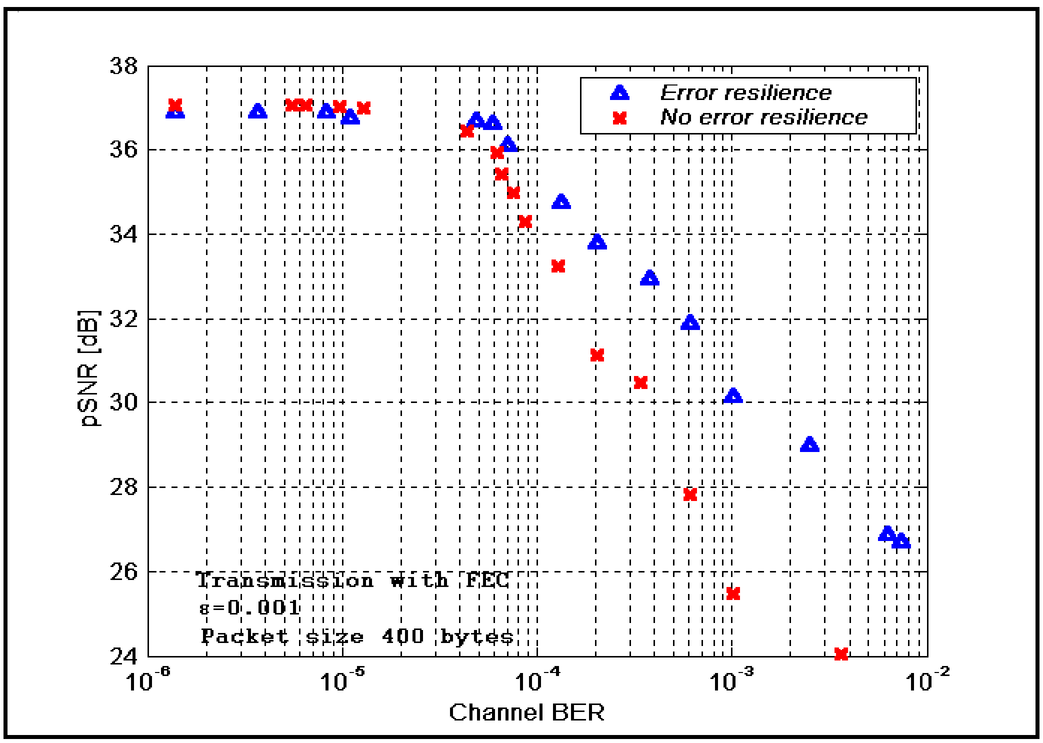


Figure B-10. Performance of JPEG2000 With and Without Error Resilient Tools. The Packet size Used for Transmission is 400 Bytes.

APPENDIX C

A. USAGE OF JPEG2000 VM8.5 SOURCE CODE

1. Compression

The following commands are used in a DOS command window in order to compress a grayscale image to a bit resolution of 2 bpp:

>*VM8_co~1 -i WOMAN.pgm -o WOMAN.jp2 -rate 2* (for Windows 98) or

>*VM8_compress -i WOMAN.pgm -o WOMAN.jp2 -rate 2* (for Windows 2000)

where *–i* identifies the image file and *–o* is the name for the compressed bitstream. The extension *.jp2* is the standard for JPEG2000. The *–rate 2* defines the bpp of the compressed bitstream; the original grayscale "*WOMAN*" image was 8 bpp. After execution of the above line we may observe that the bitstream is not exactly 2 bpp; it may be less. That happens because when we ask for a compression rate (i.e. 2 bpp), the code recognizes that as the target and tries to choose a quantization step and truncation points of EBCOT in a way that one iteration to be enough to succeed. But as we can see, it is close but not exactly the one that we asked. The VM8.5 gives the choice to choose the low tolerance in bpp (default = 0.005), higher tolerance in bpp (default = 0), the normalized base step (default = 0.0078125) and the number of iterations (default = 0). Accordingly, we can write the following line command:

>VM8_co~1 -i WOMAN.pgm -o WOMAN.jp2 -rate 2 -low_rate_tol 0.0025 -iter 10 -Cno_trunc

A list of more options is shown in Table C-1. Additional commands and further explanation can be found in the source code functions in the form of comments.

Command	Explanation
-i	Input image file with extension *.pgm*
-o	Name of the compressed bitstream
-rate	Compression ratio
-Bresync [Y|N]	Insert resynchronization markers on packet boundaries for error resilience.
-Ftiles	Breaks the image into tiles during compression. It is followed by two arguments in order to define the dimensions of the high-resolution grid.
-Frev	Apply reversible decomposition of image components.
-Flev	Define the number of decomposition levels (Default = 5 level decomposition).
-Fdecomp mallat|spacl|packet	Specifies the kind of wavelet decomposition that the code will use. A parameter follows and may be "*mallat*" (or "1"), "*spacl*" (or "21") or "*packet*" (or "321").
-Fgen_decomp <decomp string>	Specifies the general wavelet decomposition. It accepts one or more integers in the range of 0 to 3. Each string is a combination of sub-strings, which are referring to individual levels.

Table C-1. Commands of JPEG2000 VM8.5 Encoder

2. Extraction

After compressing an image we can store the bit stream in a library or send it through a channel. The command to extract the image is:

>VM8_ex~1 -i WOMAN.jp2 -o WOMAN_new.pgm

where *–i* identifies the bit stream with the extension *.jp2* . Table C-2 provides more options of VM8.5 decoder and their explanation. More detailed explanations can be found in the source code.

Command	Explanation
-o	Identifies the name of the image file after extraction.
-Cer	This command has to be used only when the compressed bitstream contains error resilience mechanism in use. Usual value for unreliable channels is 4.

Table C-2. Commands of JPEG2000 VM8.5 Decoder.

3. Example of Image Compression

Figure C-1 shows a fingerprint image. The command to compress the fingerprint image (*"finger*") is:

>VM8_co~1 -i finger.pgm -o finger.jp2 -Flev 5 -Fdecomp 2321 -Fgen_decomp 11101111111111011111

Figure C-1. Original Image 8bpp.

The resulting bit stream represents a compressed image at a bit resolution of 3.2152 bpp. The decompressed image is shown in Figure C-2.

Figure C-2. Decompressed Image.

In order to extract the image, we may use the following command:

>VM8_ex~1 -i finger.jp2 -o cmp_finger.pgm

The quality of the decompressed fingerprint image of Figure C-2 can be stated excellent not only by human observation but also by a computer matching system since the fine details are preserved much better than with image compression using JPEG. The *pSNR* of the above image is 50.754646.

4. Example of Image Compression with Region of Interest

Another option not included in Table C-1 is the region of interest generation of an image. The command " *-Rrgn xx R* (or *C*) xx_1 xx_2 xx_3 xx_4" has to be written in order to define circular or rectangular ROI. This command specifies the shape (*C* for circular and *R* for rectangular) of the ROI as well as the coordinates of the ROI on Cartesian coordinate system with origin on the upper left corner of the image and maximum value for each axis equal to one. In case of circular region we have to specify only the center of the circle and the radius again in the same coordinate system (see Figure C-3). Figure C-4 shows compression of the image *"Lena"* with rectangular (see Figure C-4 (a)) and circular (see Figure C-4 (b)) regions of interest. All the coefficients of the region of interest are shifted by a value of 20 above the background coefficients. The command that has to be used for compression at bit-rate 0.25 bpp and rectangular ROI for this image is

>vm8_co~1 -i lena.pgm -o lena_r5.jp2 -rate 0.25 -Rrgn 20 R 0.4 0.2 0.4 0.4

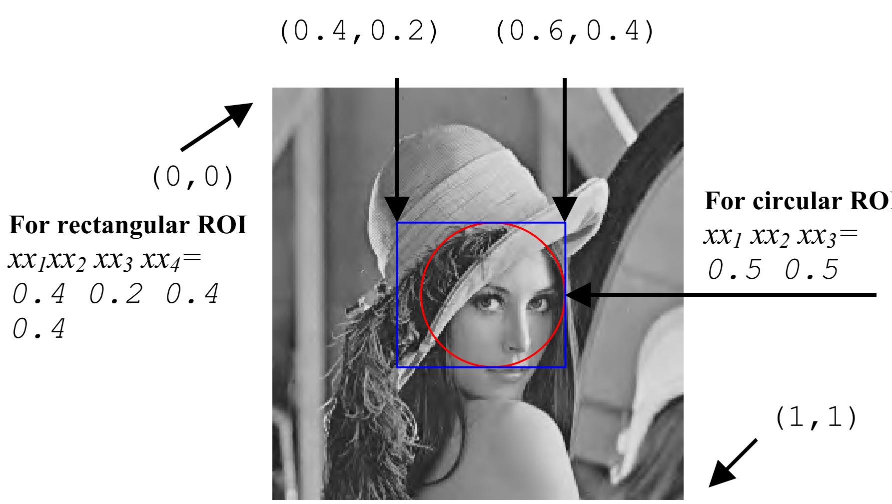


Figure C-3. Defining Rectangular or Circular ROI of an Image.

(a) Rectangular ROI (b) Circular ROI

Figure C-4. ROI Generation in a Given Image.

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

[1] A. N. Skodras, C. A. Christopoulos and T. Ebrahimi, "JPEG2000: Still Image Compression Standard" Proceedings RECPA00D 20, pp. 359-366, May 11th 2000.

[2] R. Neelamani, H. Choi and R. Baraniuk, "Wavelet-Based Deconvolution for Ill-Conditioned Systems", IEEE Trans. On Image Processing, Feb. 2000.

[3] M. S. Crouse, R. D. Nowak and R. G. Baranuik, "Wavelet-Based Statistical Signal Processing using Hidden Markov Models", IEEE Trans. On Signal Processing", Vol. 46, No. 4, April 1998.

[4] C. S. Burrus, R. A. Gopinath and H. Guo, *Introduction to Wavelet an Wavelet Transforms*, Prentice Hall, Inc, 1998.

[5] K. Sayood, *Introduction to Data Compression.* Morgan Kaufmann 1996.

[6] S. G. Mallat, "A Theory for Multiresolution Signal Decomposition. The Wavelet Representation", IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 11, No. 7, July 1989.

[7] ISO/IEC JTC1/SC29/WG1 N1271: "JPEG2000 Requirements and Profiles v5.0", March 1999.

[8] Michael D. Adams, *JPEG 2000: The Next Generation Still Image Compression Standard,* www.imagepower.com/technology/research/JPEG2000_Tutorial.pdf

[9] K. R. Rao and Y. Huh," JPEG2000", Proceedings of 2001 International Symposium of Intelligent Multimedia, Video and Speech Processing.

[10] ISO/IEC 15444-1, "JPEG2000 Image Coding System", Dec. 2000.

[11] Michael W. Marcellin, Michael J. Gormish, Ali Bilgin and Martin P.Boliek *"*An Overview of JPEG-2000"*,* Proceedings of the Data Compression Conference 2000, 523-544 Institute of Electrical and Electronics Engineers, Inc

[12] ISO/IEC FCD15444-1: 2000 (V1.0, 16 March 2000).

[13] D. Taubman, "High Performance Scalable Image Compression with EBCOT," IEEE Transactions on Image Processing, Vol. 9, No. 7, July 2000.

[14] ISO/IEC JTC1/SC29/ WG1 N1878, September. 13, 2000.

[15] Iole Moccagatta, Salma Soudagar, Jie Liang and Homer Chen, "Error-Resilient Coding in JPEG2000 and MPEG-4", IEEE Journal on Selected Areas in Communications, Vol. 18, No. 6, June 2000.

[16] Yao Wang, Jorn Ostermann and Ya-Qin Zhang, *Video Processing And Communications*. Prentice Hall 2002.

[17] C. Pian, "Study of the VM Complexity for Cost Efficient Implementation", ISO/IEC JTC1/SC29/WG1 N1471R, Dec. 3,1999.

[18] C. Christopoulos, J. Askelof and M. Larsson, "Efficient Methods for Encoding Regions of Interest in the Upcoming JPEG2000 Still Image Coding Standard", IEEE Signal Processing Letters, Sep. 2000.

[19] S. B. Wicker, *Error Control Systems for Digital Communication and Storage*. New Jersey: Practice Hall, 1995.

[20] J. G. Proakis, *Digital Communication.* McGraw-Hill, 1998.

[21] W. Stallings, *Data & Computer Communications.* Practice Hall, New Jersey 2000.

[22] F. Babich and G. Lombardi, "A Markov Model for Mobile Propagation Channel," IEEE Trans. of Vehicular Technology, Vol. 49, No. 1, January 2000.

[23] L. Wilhemsson and L. B. Milstein, "On the Effect of Imperfect Interleaving for Gilbet-Elliot Channel", IEEE Transaction on Communications, Vol. 47, No. 5, May 1999.

[24] M. Zorzi, "Some Results on Error Control for Burst-Error Channels under Delay Constraints," IEEE Trans. on Vehicular Technology, Vol. 50, No. 1, Jan. 2001.

[25] J. G. Kim and M. M. Krunz, "Bandwidth Allocation in Wireless Networks with Guaranteed Packet-Loss Performance," IEEE? ACM Trans. on Networking, Vol. 8, No. 3, June 2000.

[26] James R. Yee and Edward J. Weldon Jr., "Evaluation of the Performance of Error-Correcting Codes on a Gilbert Channel", IEEE Transactions on Communications, Vol. 43, No. 8, Aug. 1995.

[27] M. Zorsi, "Performance of FEC and ARQ Error Control in Bursty Channels under Delay Constraints", UCSD-CWC Technical Report 1998.

[28] Supavadee Aramvith and I-Ming Pao, “A Rate Control Scheme for Video Transport over Wireless Channel” IEEE Transaction on Circuit and System for Video Technology, March 2000.

[29] M.Zorzi, R. R. Rao and L. B. Milstein, “RQ Error Control for Fading Mobile Radio Channels”, IEEE. on Veh. Tech., Vol. 46, pp. 445-455, May 1997.

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Fort Belvoir, VA

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California

3. Chairman, Code EC
   Department of Electrical and Computer Engineering
   Naval Postgraduate School
   Monterey, California

4. Professor Murali Tummala, Code EC/Tu
   Department of Electrical and Computer Engineering
   Naval Postgraduate School
   Monterey, California

5. Professor Robert Ives, Code EC/Ir
   Department of Electrical and Computer Engineering
   Naval Postgraduate School
   Monterey, California

6. Dr Richard C.North, Code C855
   Spawar Systems
   San Diego, California

7. Embassy of Greece, Naval Attaché
   Washington, DC

8. Konstantinos Kamaras
   Korinthias 24 Ampelokipi
   Athens, Greece 11526